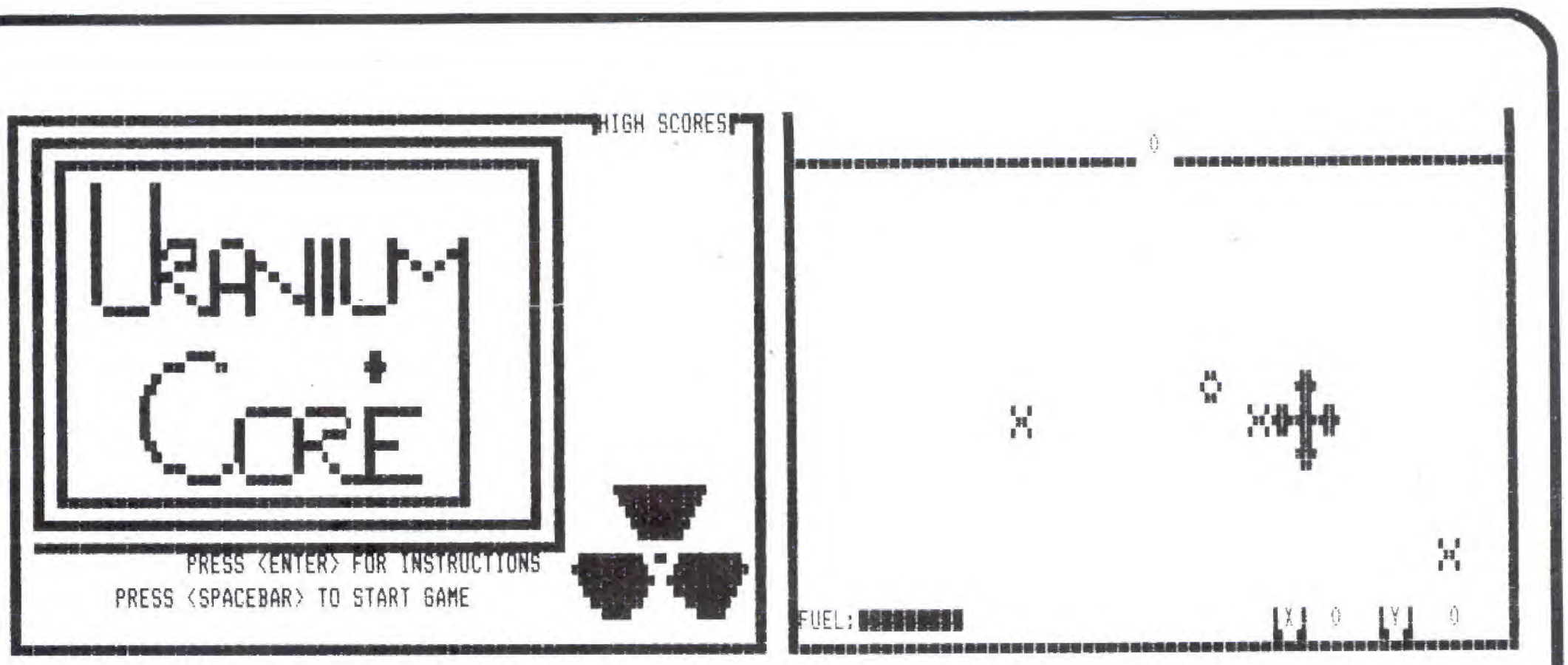
MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034. AUSTRALIA. TELEPHONE (08) 211 7244. PRICE: AUS. $2.50, N.Z. $4.00, U.K. £1.50

Registered by Australia Post — Publication No. SBQ2207

Vol. 4, Issue 1, August 1983



URANIUM CORE

Also in this issue:

PROGRAMMING:

More Arguments for Machine Language Subroutines

True or False

REVIEWS:

Typing Tutor

Pyramid 2000

HARDWARE:

The Dreaded Reboot

SOFTWARE:

- Aristocrat — Colour
- Compound Multiplication and Long Division Level I
- DEFUSR Function — Level II
- Single Key Menu — Model 3
- Starshoot — Level I
- Startrek — Colour

• TRS-80 • SYSTEM 80 • VIDEO GENIE
• PMC-80 • HITACHI PEACH
• TRS-80 COLOUR COMPUTER

***** ABOUT MICRO-80 *****

EDITOR:	RYSZARD WIWATOWSKI
ASSOCIATE EDITORS:	
SOFTWARE :	CHARLIE BARTLETT
HARDWARE :	EDWIN PAAY

MICRO-80 is an international magazine devoted to the Tandy TRS-80 Model I, Model III and Colour microcomputers, the Dick Smith System 80/Video Genie and the Hitachi Peach. It is available at the following prices:

	12 MONTH SUB.	SINGLE COPY
MAGAZINE ONLY	$ 26-00	$ 2-50
CASSETTE PLUS MAGAZINE	$ 65-00	$ 4-00 (cass. only)
DISK PLUS MAGAZINE	$125-00	$10-00 (disk only)

MICRO-80 is available in the United Kingdom from:
U.K. SUBSCRIPTION DEPT. 24 Woodhill Park, Pembury, Tunbridge Wells, KENT TN2 4NW

MAGAZINE ONLY	£ 16-00	£ 1-50
CASSETTE PLUS MAGAZINE	£ 43-60	£ N/A
DISK PLUS MAGAZINE	£ 75-00	£ N/A

MICRO-80 is available in New Zealand from:

MICRO PROCESSOR SERVICES, 940A Columbo Street, CHRISTCHURCH 1 N.Z. Ph 62894

MAGAZINE ONLY	NZ$ 43-00	NZ$ 4-00
CASSETTE PLUS MAGAZINE	NZ$ 89-00	NZ$ 5-00
DISK PLUS MAGAZINE	NZ$175-00	NZ$15-00

MICRO-80 is despatched from Australia by airmail to other countries at the following rates:

(12 MONTH SUB)	MAGAZINE	CASS + MAG	DISK + MAG
PAPUA NEW GUINEA	Aus$40-00	Aus$ 83-00	Aus$ 143-00
HONG KONG/SINGAPORE	Aus$44-00	Aus$ 88-00	Aus$ 148-00
INDIA/JAPAN	Aus$49-00	Aus$ 95-00	Aus$ 155-00
USA/MIDDLE EAST/CANADA	Aus$55-00	Aus$102-00	Aus$ 162-00

Special bulk purchase rates are also available to computer shops etc. Please use the form in this issue to order your copy or subscription.

The purpose of MICRO-80 is to publish software and other information to help you get the most from your TRS-80, System 80/Video Genie or Peach and its peripherals. MICRO-80 is in no way connected with any of the Tandy, Dick Smith or Hitachi organisations.

** WE WILL PAY YOU TO PUBLISH YOUR PROGRAMS **
Most of the information we publish is provided by our readers, to whom we pay royalties. An application form containing full details of how you can use your microcomputer to earn some extra income is included in every issue.

** CONTENT **
Each month we publish at least one applications program in BASIC for each of the microcomputers we support. We also publish Utility programs in BASIC and Machine Language. We publish articles on hardware modifications, constructional articles for useful peripherals, articles on programming techniques both in Assembly Language and BASIC, new product reviews for both hardware and software and we print letters to the Editor.

** COPYRIGHT **
All the material published in this magazine is under copyright. This means that you must not copy it, except for your own use. This applies to photocopying the magazine itself or making copies of programs on tape or disk.

** LIABILITY **
The programs and other articles in MICRO-80 are published in good faith and we do our utmost to ensure that they function as described. However, no liability can be accepted for the failure of any program or other article to function satisfactorily or for any consequential damages arising from their use for any purpose whatsoever.

***** CONTENTS *****

		PAGE
REGULARS		
EDITORIAL		2
INPUT/OUPUT - LETTERS TO THE EDITOR		2
MICROBUGS		10
USERS GROUPS		10
DEPARTMENTS		
KALEIDOSCOPE		3
PEACH BOWL		4
GROUP ONE		5
FORM THREE		5
PROGRAMMING		
TRUE OR FALSE?		6
MORE ARGUMENTS FOR MACHINE LANGUAGE SUBROUTINES		7
REVIEWS		
PYRAMID 2000		8
TYPING TUTOR		9
HARDWARE		
THE DREADED REBOOT		9
SOFTWARE		
STARTREK	CC	12 & 18
STARTREK	PEACH	12 & 22
ARISTOCRAT	CC	12 & 21
ARISTOCRAT	PEACH	12 & 25
URANIUM CORE	L2/16K	12 & 26
SINGLE KEY MENU	M3DISK	13 & 28
DEFUSR	L2/4K	15 & 30
COMPOUND MULTIPLICATION AND LONG DIVISION	L2/4K	16 & 31
STARSHOOT	L1/4K	17 & 32
MICRO-80 PRODUCTS CATALOGUE		CENTRE
INDEX TO VOLUME 3		33
NEXT MONTH'S ISSUE		35
CASSETTE/DISK EDITION INDEX		36
ORDER FORM		36

MICRO-80 IS REGISTERED BY AUSTRALIA POST - PUBLICATION NO. SBQ2207

AUSTRALIAN OFFICE AND EDITOR: MICRO-80, P.O. BOX 213, GOODWOOD, SOUTH AUSTRALIA, 5034, TEL. (08) 211 7244

U.K. SUBSCRIPTION DEPARTMENT: 24 WOODHILL PARK, PEMBURY, TUNBRIDGE WELLS, KENT TN2 4NW

PRINTED BY: SHOVEL & BULL PRINTERS, 379 SOUTH ROAD, MILE END, SA., 5031

PUBLISHED IN AUSTRALIA BY: MICRO-80, 433 MORPHETT STREET, ADELAIDE, S.A., 5000

***** EDITORIAL *****

As we begin volume four of MICRO-80, a number of profound changes have taken place in the area of personal computers of particular interest to our readers. Production of the TRS-80 Model I stopped long ago and the System-80/Video Genie is also no longer being made. The introduction of the Models 4 and 12 will no doubt see an end to the production of the Models 2 and 3. But despite this, the most popular computer in the Australian home is still the Model I or System-80 and the majority of our readers own 16K Level 2 cassette-based systems. Disk users and Model 3 owners comprise a large proportion of the remainder followed by a smaller number of Colour Computer and Peach owners. What then can our readers expect from MICRO-80 in the coming months?

Firstly, a wide variety of programs to run on your computer including games, utilities and applications. More articles on programming techniques, hardware projects and topics of special interest such as the technical aspects of disk drives and operating systems, at various levels to suit the beginner and the more advanced user. Critical reviews of both software and hardware products available for your computer and as many hints, tips and useful facts to help you learn more about your computer as we can squeeze into the magazine. As a measure of our sincerity, we are putting the finishing touches on a free software offer to all our new subscribers and those renewing their subscriptions (starting with this issue). For our Group One and Form Three readers, the Software Library has been revised and improved with the addition of several new programs. A completely new Software Library has been developed for our Colour Computer readers and Peach owners will get a choice of one of three commercially available games!

BUGET COMPUTERS

At the moment, the interest of computer manufacturers is focussed on the potentially very large domestic market. Although this section of the community is eager to learn more about the personal computer, it is also very reluctant to outlay too much money to buy one. Therefore, the recent trend has been to offer a very low cost entry into computing like the VZ-200 priced at $199. So it is not surprising to see Tandy introduce the MC-10 Colour Computer into the U.S. market for just under $US120.

A smaller, less powerful machine than the TRS-80CC, the MC-10's main features are its compact size, colour graphics and sound. Inside it features the Motorola MC6803 processor, 4K of RAM (which can be expanded externally by an extra 16K) and an 8K BASIC interpreter in ROM. Although the cassette and RS232 ports maintain hardware compatibility with the TRS-80CC, the different Microcolour BASIC and the different processor prevent direct software compatibility. I expect that when released in Australia, the price of the MC-10 will be comparable with that of the VZ-200.

Unfortunately, the advantages of low-price are offset by the general lack of good quality software. No matter how impressive the hardware, without software it is of little use to anyone. It will also be interesting to see the price of software for these machines - at around $25 per game, it would require the purchase of only 8 games before you exceed the price of the machine itself!

- 0000000000 -

***** INPUT/OUTPUT *****

From: M.J. Moore - Oxley, Qld.

I would appreciate your publishing this plea for help in the "Input/Output" section of your magazine in the hope that a reader may have overcome the problem that I have been experiencing.

I have a System 80 to which I have attached a TC8 Cassette Operating System via a Syspand 80 bus converter. The system will only enable switching of the remote cassette motor and not transfer of programs or data. The TC8 has been tested and operates perfectly on the TRS-80. I removed the Syspand 80 and connected the TC8 directly to the System 80, but the result was the same.

It would appear that I have a software problem with the driver routine for the TC8. Can anyone please help?

(At a guess, I would say your problem is related to the cassette port differences between the System 80 and the TRS-80, particularly in the treatment of bit 2 (see letter p.5 July '82 issue). Perhaps one of our readers can provide a solution to this problem - Ed.)

From: P. Bunyan - Jervis Bay, A.C.T.

I am in the process of building the joystick interface controller and was wondering if the interface can be used as a Centronics parallel interface for a printer or a serial interface. I think it would be good to have an article about this in MICRO-80. In the meantime, could you give me some information as to how the joystick interface could be used to control a printer?

(I presume that you are referring to the "Joystick and I/O Ports" project by Allan Dent, in

which case the answer is 'No'. This joystick interface is not suited to such an application and Allan doesn't recommend trying to use it in this way. However, I agree that it would be good to have a constructional article on a Centroncis printer interface in the magazine. Any takers? - Ed.)

From: K.W. GLasson - Karalee, Qld.

I have noticed from time to time in MICRO-80 a few one line programs and I thought you may be interested in one of my "one liners". I call it a Mosaic Generator for reasons which are obvious once you've run it. It helps some people to relax just as you can do by staring at an open fire or by watching fish swimming around an aquarium.

Might I add a suggestion that when the time lapse between your acceptance of a program and its appearing in MICRO-80 exceeds say 6 months, that the contributor be approached for any new information concerning the program prior to its going to press. In my own case, I submitted a program called "Loan Calculation Package" and its acceptance for publication was a source of pride. However, its appearance in MICRO-80 some 18 months later was a source of embarrassment. During the interim I had improved it in many ways and your magazine could have had the improved version for the asking.

Here is the "one liner" -

```
10  CLS: FOR S = 1 TO 5: FOR D = 1 TO 100: X = RND(63): Y = RND(23): SET (X,Y):
    SET (127-X,Y): SET (X,47-Y): SET (127-X,47-Y): NEXT D: FOR F = 1 TO 500: NEXT F:
    NEXT S: GOTO 10
```

I hope you like it.

(Thank you for your suggestion and I'm sure our readers will find the one-liner interesting. Although what you propose has merit, we believe that it would not be practical to implement. Most, if not all, of the programs that we have on file are at least six months old and we would need to contact the authors for every program to be published. The updated version (if any was available) would have to be tested and the accompanying description altered where required. Assuming that no problems arise, this procedure would at least double the amount of time spent in the preparation of each program and inevitably lead to production delays - an intolerable situation. -Ed.)

From: Gavin Daniels

I have recently purchased an expansion unit and disk drives for my System 80 MK II and I am finding it difficult to put all my system programs from tape to disk. Could you please help remedy this problem by specifying a particular program which can do it, or is it possible to change the listing of the System Copier on the MICRO-80 cassette so it will save to a disk?

Is it also possible to change all my machine language programs like ASYLUM which save data to cassette to save it to disk? If so, could you please send me the modifications for the following programs: ASYLUM, ADVENTURE 9, EDITOR ASSEMBLER PLUS, ZMON, CUBE, HOUSEHOLD ACCOUNTS VERS 3.0, HELLFIRE WARRIOR, BMON, ROBOT ATTACK, GALACTIC EMPIRE, etc.

Please note that the games GALACTIC EMPIRE (Broederbund software) and HELLFIRE WARRIOR (Dunjonquest) are written in BASIC.

(Disk operating systems, like DOSPLUS and NEWDOS80, provide utilities for exactly this purpose, viz. TAPE and LMOFFSET respectively. These do not, however, relocate machine language programs in the true sense of the word. They rather displace the program so that it can be LOADed from DOS and add an appendage that is executed first and which moves the program to its correct location in memory before passing control to it.

Modifying machine language programs intended for cassette systems so that they save and load data from disk is possible, but not recommended. A much more expedient method is to buy the disk versions of these programs from the original vendor. To save and load data from BASIC programs is somewhat easier but depends on the given program. The Software Library booklet lists modifications for the disk version of Household Accounts as an example of how it can be done. Note that there is a correction to these in Microbugs (August '82 issue). -Ed.)

- 0000000000 -

DEPARTMENTS

***** KALEIDOSCOPE *****

This month's programs for the Colour Computer (Aristocrat and Startrek) show some ways to get around the limiting text display. Although we have received one or two programs from our readers,

we hope to see a lot more in the near future.

Here are some more hints you may find helpful on your Colour Computer.

Don't be alarmed or discouraged if your program produces a syntax error for no apparent reason. Evidently, there is a bug in the BASIC ROM that sometimes causes a program using the PCLEAR statement to be stopped with a syntax error the first time it is RUN. The cure is simply to RUN it again. If you get another syntax error, then you'd better look closely at the line in question.

Although most of the machines in Australia should contain Version 1.1. ROMs, some very early or imported machines may contain Version 1.0 ROMs. There are rumours that the latest machines in the U.S. have 1.2 ROMs and there may be a few here in Australia. To check exactly which ROMs your Colour Computer contains, type 'EXEC 41175 (Enter)'.

If you find the green background colour a bit hard on the eyes then try the following:

POKE 359,57 : SCREEN 0,1 (Enter)

This changes the background colour to orange reasonably permanently. However, if for some reason your computer appears to lock up, then type (even if it doesn't appear on the screen):

POKE 359,126 (Enter)

What has happened is that the screen displays the graphics page and doesn't return to the text page as it would under normal circumstances.

- 0000000000 -

***** PEACH BOWL *****

Two more programs for our Peach users this month - Aristocrat and Startrek. Originally submitted for the Model 1, they have been modified to run on the Peach. However, a few of our Peach readers have sent in programs and we expect to publish some of these next month.

In September 1982 issue, we published a program for our disk users called Sector Editor. Although this program is a very useful utility with which to explore and repair disks, it has the drawback that the BASIC screen display routines are very slow. Well, one local user by the name of Geoff Drury developed a modification to the original program to speed up the displaying of information. What takes up a lot of time is the conversion of the binary sector data into displayable ASCII data and if this can be done by a machine language subroutine rather than in BASIC, the process is sped up enormously. Below is a list of the modification. Try it and see the difference.

```
50 SCREENO,,0: CLEAR 4096+128:MCODE$=STR
ING$(120,"#")
52 MCADDR=256*PEEK(VARPTR(MCODE$)+1)+PEE
K(VARPTR(MCODE$)+2):ADR=MCADDR
54 READ B$:IFB$<>"END" THENPOKE ADR,VAL(
"&H"+B$):ADR=ADR+1:GOTO54
56 WIDTH 80:CLS:DIM SECTOR$(16%):' SECTO
R$() - USED IN THE SECTOR COPY
   ROUTINE - SEE LINE 1000
90 PRINT "Function "; : CMD$=INPUT$(1) :
 PRINT CMD$ : ON INSTR(1%,"DZCQ",CMD$)
   GOTO 110,860,980,1370
1999 '

     '           PRINT/DISPLAY SECTOR DAT
A
     '
2000 CODEADDRH=PEEK(VARPTR(SECT$)+1):COD
EADDRL=PEEK(VARPTR(SECT$)+2)
2010 POKE MCADDR+&H63,CODEADDRH : POKE M
CADDR+&H64,CODEADDRL
2020 EXEC MCADDR
60100 DATA 6F,8D,00,5E,AE,8D,00,5B,1F,10
,C4,0F,1F,98,34,16,A6,8D,00,4E
60110 DATA 48,48,48,48,BD,DD,32,C6,04,8D
,3C,35,16,10,8E,00,10,A6,80,34
60120 DATA 36,BD,DD,32,BD,B1,16,35,36,31
,3F,26,F0,34,36,C6,06,8D,20,35
60130 DATA 36,30,10,C6,0F,34,36,BD,FA,73
,BD,B0,BE,35,36,30,88,10,6C,8D
60140 DATA 00,10,A6,8D,00,0C,81,08,25,AE
,39,BD,B1,16,5A,26,FA,39,0,0,0
60150 DATA END
```

DELETE LINES 2000 TO 2140

- 0000000000 -

***** GROUP ONE *****

For our Level 1 users this month we have Starshoot and Compound Multiplication and Long Division, while for our Level 2 readers we present Uranium Core and a utility to provide the DEFUSR statement in non-disk systems.

Reader John Smith has sent in these helpful hints:

In some games and competitive computer activities it is useful to have a limited time for input. This facility is not available as a BASIC function in the TRS-80 machines, although it is available in Assembly on the Model II. This latter doesn't help people like me who haven't yet learned how to use such straing symbolism (!), and is no help to Model I and Model III owners, who, I believe, have no such facility. So I wrote the following BASIC routine to solve the problem. It is quote short and simple. Most of the listing here is REMarks, making the listing self-explanatory. I trust it may interest your readers.

```
100 '          ***** "TIMETRY" *****
110 '  J.D.SMITH, HAWTHORN, SOUTH AUSTRALIA
120 '      <<<<<  6th. March, 1982. >>>>>
130 'This is a means of limiting the time allowed for input to a programme.
140 '
150 REM >>>NOTE<<<: In Level II BASIC the apostrophe (') may be used
160 '    instead of the word "REM".  For Level I you have to do a bit more typing
!
170 '
180 '**********Set the parameters**********
190 '
200 SEC = 10 'Time in seconds allowed for an input to be made.
210 LM=4 'LM is the limit on "no entry" loops.
220 TD=SEC*46 'TD is the time delay count.  Multiple will depend on
230 '           clockspeed of computer.  This suits Models I & III.
240 '          For Model II use a factor of 100 instead of 46.
250 TC=0 'TC is the count of cycles with no entry at all.
260 '
270 '**********This is the time-input loop**********
280 '
290 PRINT"YOU HAVE" SEC "SECONDS TO ENTER A NUMBER OR WORD"
300 TM=0:A$=""
310 I$=INKEY$:TM=TM+1:IF TC=LM THEN 380ELSE IF TM=TD THEN 370ELSE IF I$=""THEN 3
10
320 IF I$=CHR$(13)THEN 470'Note:CHR$(13) is <ENTER>.
330 PRINT I$;
340 A$=A$+I$ 'A$ is used to build up the input string.
350 ' A string is needed for INKEY$.  For numerical input use VAL(A$).
360 GOTO 310
370 TC=TC+1:PRINT"TOO SLOW":GOTO 300
380 PRINT"TOO LONG. TRY SOMETHING ELSE!":GOTO 520
390 '
400 '**********End of input loop**********
410 '
420 'It is good programming  practice to have an "END" statement only
430 ' at the actual end of the listing.  Some computers do not accept
440 ' programme entry  or will not list after an "end", so it is good
450 ' practice to  either  use a  "GOTO"  as in  line 360  above,  or
460 ' to use "STOP" to terminate elsewhere than at the last statement.
470 PRINT:PRINT "A$=" A$
480 PRINT "VAL(A$)=";VAL(A$)
490 A=VAL(A$)
500 PRINT"VAL(A$)+2=";A+2:PRINT
510 GOTO 250
520 END
```

- 0000000000 -

***** FORM THREE *****

Our Model 3 cassette users should protect high memory and move the DEFUSR routine there. Exclusively for our Model 3 NEWDOS users we present: - Single Key Menu. However, with a little bit of hunting around, I suspect users of other DOS's should be able to adapt it to their system.

A word of caution regarding this last program:-

Changes are made to the Operating System's two most important modules, SYS0 and SYS1. These

should be checked very carefully. These should be checked very carefully. Take note of the author's suggestion and work with a backup copy of your system disk. Also, it may happen that the vendor of the DOS may supply corrective patches that will conflict with these changes sometime in the future. For your benefit record the changes made and keep it in the DOS manual for future reference.

Finally, I wish to draw the attention of our Model 3 cassette subscribers to the Microbug referring to the Movie utility. I hope this has not caused too much inconvenience.

- 0000000000 -

PROGRAMMING

***** TRUE OR FALSE? *****

by Noel Rossiter

Power up your trusty TRS-80 Level 2, and when all is READY key in:

```
?(X=Ø)
```

and quick as a flash, comes the reply:

```
-1
```

Strange? Well, now try:

```
?(X=-1)
```

to which, of course, you get the answer:

```
Ø
```

To complete the exercise, key in and run the following:

```
FOR I=1 TO 1Ø:X=(X=Ø):?X;:NEXT
```

and you will get:

```
-1 Ø -1 Ø -1  Ø -1  Ø -1 Ø
```

What you have been doing here is printing the two possible values of the truth function. Enclosing a statement like X=Ø in brackets to make (X=Ø) is an indication to Level 2 BASIC that the truth of the statement is to be tested, and the appropriate value of the truth function returned; as indicated already, the truth function can only have one of two values, -1 if the statement tested is true, and Ø if it is false.

The statement X=(X=Ø) causes X to change its value from Ø to -1 and the reverse each time it is executed, because if X is equal to Ø the expression in brackets is true (truth value = -1) so that -1 is returned and X is put equal to that value; on the other hand, if X does NOT equal Ø the truth function is returned with a value of Ø and X is set to that value. This, of course, can be used to make a flip-flop switch in programs where particular program sequences have to be executed alternately; typically:

```
 ......................
 90 X=(X=Ø):IF X=Ø THEN 200
100.....................
 ......................
190 GOTO 90
200 ....................
 ......................
290 GOTO 90
```

There are other convenient uses of the truth function; try your machine on:

```
X=Ø:FOR I=1 TO1Ø:X=X-(I<6)+(I>5):?X;:NEXT
```

You will, by now, be able to figure out that the response will be:

```
1 2 3 4 5 4 3 2 1 Ø
```

Since the truth function can be multiplied and divided, as well as being added and subtracted (Y=5*(X=3)) will return Y with a value of Ø unless X=3 in which case Y will have a value of -5) the truth function offers an elegant alternative to the use of READ and DATA statements to set values on a program. For example, suppose the 27 values of a variable X(I) were to be:

```
1   2   3   4  .......9
10  20  30  ..........90
100 200 300 ..........900
```

(Don't ask me why), you could set up a DATA statement containing the 27 values and use:

```
FOR I=1 TO 27:READ X(I):NEXT
```

Alternatively, you could omit the DATA statement, and use:

```
FOR I=1 TO 27:X(I)=I-9*(I-1Ø)*(I>1Ø)-9Ø*(I-19)*(I>19):NEXT
```

You can also use the truth function to clip off a variable at maximum and/or minimum values; for example:

```
IF K>6Ø THEN J=6Ø ELSE IF K<Ø THEN J=Ø ELSE J=K
```

allows J to range between Ø and 6Ø according to the value of K; so does:

```
J=K+(K-6Ø)*(K>6Ø)+K*(K<Ø)
```

Finally, you can avoid the /Ø error by use of the truth function:

```
P=Q/(R-(R=Ø)) may return a wrong value of P if R=Ø
```

but P=Q/R would cause the program to bomb out.

I have not attempted to exemplify all the places in which it may be useful or convenient to make use of the truth function; I have only listed a few cases where I have found it useful. I must admit that minus sign complicates things a bit, and I always have to take a deep breath and work things out carefully. I understand that Level 1 BASIC returns a truth value of 1 if true, and this would make life a little easier. However, the Level 2 arrangement has the benefit of being consistent with the NOT logical operator, because NOT Ø=-1 and NOT-1=Ø. I guess we can't have everything.

- 0000000000 -

***** MORE ARGUMENTS FOR MACHINE LANGUAGE SUBROUTINES *****

by S.H. Liggins

To speed up my BASIC Programs I had often considered including machine language routines, but I didn't like the usual methods of doing it. I wanted the routines to be entirely relocatable and suit machines of any size. I didn't want to worry about setting Memory size, or other fiddling with BASIC pointers. I wanted to be able to include several such routines in the same program without worrying which was where. At times, I also needed to pass more than the single argument allowed by BASIC.

So I decided to store them in integer arrays. The main disadvantage is that they are slightly more tedious to use, as it is necessary to ensure that BASIC doesn't do any sneaky relocation of variables at critical times.

An example is the "BEEP" which I use for error warnings in data entry programs.

INITIALISATION:

```
AG% = Ø : AD% = Ø : RE% = Ø                 Argument, address, result variables
READ I : DIM BP%(I)                         Define array to store the routine
FOR J = Ø TO I
    READ BP%(J)                             Store the routine
NEXT J

DATA  30,4120,32717,15882,2049,9038,9054,-9755,-9759,1816
DATA  32717,-20736,19720,31068,-14153,-18565,16072,17154,
DATA -43,-496,8253,3576,-3296,-18680,456,-1,12321,2304,
DATA -712,6361,-52,-1
```

SUBROUTINE:

```
(AG% already set up with frequency * 256 + duration)
AD% = VARPTR(BP%(0)) : DEFUSR=AD% : RE% = USR(AG%) : RETURN
```

From the above the persevering reader will notice that the DATA statements contain 2 byte integer values to be read into array BP% rather than 1 byte values to be POKEd into memory.

It is important that no variables be created between the DEFUSR statement and the USR statement. I created all the variables in the initialisation routines to be on the safe side.

A second example is the familiar 'store/restore screen display' problem. It is a simple bulk move, requiring three arguments, source address, destination address, and number of bytes. Each argument is stored in the appropriate part of the MV% array.

INITIALISATION:

```
VD% = 15360 : AD% = Ø : RE% = Ø              Video addr, M/L addr., result variables
DIM ST%(512)                                 Storage for screen (= 1024 bytes)
READ I : DIM MV%(I)                          Define array to store the subroutine
```

```
FOR J = Ø TO I
    READ MV%(J)                                  Store the subroutine
NEXT J
DATA  0,8448,0,4352,0,256,0,-4664,-13904
STORE SCREEN:
MV%(1) = VD%                                     Source
MV%(3) = VARPTR(ST%(0))                          Destination
MV%(5) = 1024                                    Length
AD% = VARPTR(MV%(0)) : DEFUSR = AD% : RE% = USR(0) : RETURN
RE-DISPLAY SCREEN:
MV%(1) = VARPTR(ST%(0))                          Source
MV%(3) = VD%                                     Destination
MV%(5) = 1024                                    Length
AD% = VARPTR(MV%(0)) : DEFUSR = AD% : RE% = USR(0) : RETURN
```

Curious readers will be wondering why the arguments are stored in MV%(1),MV%(3) and MV%(5). This is because the actual machine language code is as follows:

```
NOP       LD HL, source
ØØ          21    ssss
MV%(Ø).....       MV%(1)

NOP       LD DE, destination
ØØ          11    dddd
MV%(2).....       MV%(3)

NOP       LD BC, length
ØØ          Ø1    1111
MC%(4).....       MV%(5)

LD A,B   OR C   RET Z    LDIR.....    RET
78        B1      C8      ED     BØ    C9
MV%(6).....    MV%(7).....       MV%(8).....
```

Note: Non-disk users will have to POKE AD% into 16526,16527 as usual instead of DEFUSR = AD%

- 0000000000 -

REVIEWS

***** PYRAMID 2000 BY RADIO SHACK *****

A Review by Brian J. Fillery.

I had read a couple of reviews of Pyramid before I got the chance to really try it out myself. The first was written by someone who obviously did not like adventure games, and he damned it out of hand. The second review was kinder and gave me hope. Needless to say, neither of these reviews were in MICRO-80 nor were they Australian.

I can get as frustrated as the next man, so I was wondering what I would do with Pyramid. Well, I did get frustrated, but isn't that what the game is about? Truthfully, it is a difficult game for those with no ingenuity. You have to use your brain to do things, to work it out.

When the game starts, you find yourself outside an entrance to a Pyramid (where else?). You have to communicate with your Astral self, they say in the large book that comes with the game. It is large in size, but tells you very little of use. Hence, use your brain!

Whilst I was playing the game, I felt hampered talking in two-word sentences to my Astral self so I decided that it was a rather dim robot instead. This suited my personality.

The first thing that comes up is a message. "Welcome to Pyramid"...then nothing happens. In desperation one presses ENTER and the game starts. You instruct your robot/astral self to 'go in' and from there on, you are on your own. There are serpents, gold nuggets, rooms with hieroglyphics, bottomless pits, but you use your wits to get you round. If you have any sense, you will make a map of your various turns, so you know where you are.

There are various things you can do with two-word sentences. You can go, get, drop, look, take, climb, throw, etc. If you're not sure what you have collected, you can take 'inventory' and 'score' will tell you how badly you are doing.

There are apparently about 30 rooms and one 10 room maze. I didn't get much further than about 6 rooms, but time was not on my side and I didn't make a map, which was stupid. Was it my imagin-

ation or did the rooms change round the second time I played it?

If you are desperate you could try peeking into the game to find out how it works and what all the right words are, but then what is the point of playing it?

This is an adventure game. It is hard for some and easy for others, but it will certainly keep you entertained for a long time to come.

- 0000000000 -

***** MICROSOFT TYPING TUTOR *****

A Review by A.F.J. Bell

This cassette based program requires 16K, and comes from the producers of the BASIC used in the TRS-80, and of such great programs as EDTASM Plus and ZBUG. My copy cost $19.95.

It comes with an excellent 20 page manual which simply and clearly explains what the program does and why, how to set the program up, and how to use it. It also has some hints on good typing.

At the beginning of the program one is offered the options of learning letters, numbers, or symbols. After the computer sets itself up, one is offered the option of "Typing Tutor" or "Practice Paragraph".

If one chooses Typing Tutor, two blocks of four randomly chosen characters will be displayed on the screen with the cursor underneath showing that one has to type the indicated lesson. The lessons are set out so that the easiest characters are practised first. At the end of the blocks a new lesson is generated, depending on the results of the previous lesson. That is, if one is typing a given character accurately and quickly, then it is removed and replaced by another character. After 10 lessons the computer gives a progress report, and then allows the options of slower, same, or faster response, or of Practice Paragraph. If Practice Paragraph is chosen, the computer will generate a several line paragraph using only the characters already practised, and one types the characters displayed just as in the Typing Tutor option. At the end of the Practice Paragraph one's results are analysed and displayed, and the options of Typing Tutor and Practice Paragraph are again allowed.

I found this an easily understood and enjoyable to use program, and had no difficulty putting it on ESF wafer. Better still, it helped me to increase my speed and accuracy. I strongly recommend it.

- 0000000000 -

HARDWARE

***** THE DREADED REBOOT *****

by R.G. Burgin

I wonder how many of us have been typing away on a Model I TRS-80 only to suddenly have the disk drive start up and have the whole system reboot for no apparent reason? Of course, there is always a reason. It is just that us mere mortals have a little more trouble finding it than the computer gods who write the magazines but fail to give us answers we can access.

My system on 48K, LNW expansion, MPI B92, and Dick Smith Printer under NEWDOS 80 V2 gradually developed the dreaded reboot to the point of being almost unusable. I believe the problem is almost unknown if the keyboard is used alone. It certainly never happened to me when I only had 16K. It seems obvious that the more that is hung onto the system, the more prone it is to trouble. Well, Tandy did introduce the Model III for some reason, didn't they? I don't think they really care about RFI or the Model I wouldn't be, would it?

OK, the increased number of bits and pieces in an expanded system will unfortunately add to the number of spontaneous reboots; after all, there's more for the odd cosmic ray to hit. It only takes one bit in the 48K of RAM to be a little doubtful and bingo! Even if it doesn't reboot, the CPU is off in never never land, ignoring the keyboard.

You and I can't do anything about the acts of God like the above, but you sure can take the steps I took to ease my problems. The first area to look at is the push on edge card connectors tying the system together. Good practice says that all the contacts should be gold-plated to prevent corrosion. After all, we are dealing with radio frequency currents at very low power levels so corrosion is almost guaranteed. So why didn't Tandy gold-plate the connectors? It wouldn't be too expensive when the PCB was bare. After all, the plug makers can do it and their prices aren't too bad.

You can pull everything to bits and clean the contacts with a pencil eraser - it works, for a while. Don't forget the disk drive and printer connectors. KEEP OUT of the disk drive - its makers weren't so frugal - the contacts here are gold-plated. If yours aren't, have a good think about what the rubbings from the eraser could do inside before having a go.

If you read any of the American magazines on computing, no doubt you will have seen methods of gold-plating the contacts of existing PCB's. Forget it! To buy the chemical needed would cost in the region of $400.00 in Australia plus some gold. I know, I spent a small fortune in phone calls to find out.

There is advertised in the American magazines another fix called Gold Plug 80 for about $10 US per plug for which I very nearly sent off the money. But would I get it sent to a foreign country, and even if it is a reputable company, how long would it take? The solution, however, is a lot nearer to home than that. Strangely (? - Ed.) it was MICRO-80 themselves who solved my problem. I was in their office having a quiet snoop and decided to have a bit of a grizzle about the problem when they suggested I go and see Grantham Pty. Ltd., in Gilles St. Adelaide.

These good people produced a connector strip of 80 gold flashed contacts which can be cut off at the desired length and soldered to the existing contracts. The strip is coded S-700 QQ 170 and was less than $25 when I got mine. I have had no further reboots caused by bad cable connectors since fitting these gold contacts. Power line surges have been the cause of some further problems but if the Editor is willing, we might be able to discuss that another time.

(Certainly. By the way, Grantham Pty. Ltd. is now known as Magmedia at the same address - Ed.)

- 0000000000 -

***** MICROBUGS *****

Although we make every effort to ensure accuracy in the material we publish, inevitably errors and omissions will occur. In this section, we print corrections to those bugs that have been reported.

CHECKSUM Vol. 3, No. 11 October 1982 pp.31-32.

The source listing printed in the magazine is for the disk version of the program. The Level 2 version should have the following changes made:

(1) in line 250, the Origin should be 41E2H.

(2) in line 1000,the Origin should be 428AH.

MOVIE FOR MODEL 3.

The modified version of MOVIE for the Model 3 that appeared on the October 1982 cassette was not the final version and does not work on the Model 3. This month the cassette includes the correct copy of MOVIE for Model 3 users. The October 1982 disk contained the final version and is not affected by this correction.

- 0000000000 -

***** USERS' GROUPS *****

The following is a list of User's groups of which we are aware. Many are interested in a wide variety of computers. For further information about any of the groups, contact the person indicated. If you have a group not mentioned here, please let us know so that it can be included in this section.

** AUSTRALIA **

AUSTRALIAN CAPITAL TERRITORY

CANBERRA GROUP

Cont: Bill Cushing
10 Urambi Village, Kambah, 2902.
Meet: 3rd Thurs. monthly, 7.30pm
Urambi Village Comm. Centre.
Crozier Circuit, Kambah.

NEW SOUTH WALES

BLUE MOUNTAINS OF N.S.W.

Cont: Greg Baulman
Tel: Home (047) 51 3221
Meet: 1st Fri. monthly, 7.30pm
Springwood Civic Centre.

NORTHERN & WESTERN SUBURBS COMPUTER USERS GROUP

Cont: David Coupe, Tel: (03) 370 9590
Meet: CPM Data Systems, 284 Union Rd.
Moonee Ponds. Alt. Thurs. 7pm.

PENINSULA COMPUTER GROUP

Cont: George Thompson, 3 Patterson St.
Bonbeach, 3196. Tel: 772 2674.
Meet: 2nd Tues. monthly, Chisholm College,
Frankston.

TRS-80 SYDNEY EASTERN SUBURBS USERS GROUP

MAPPER CP/M USERS GROUP

Cont: Dan Lawrence
G.P.O. Box 2551, Sydney, 2001.

WOLLONGONG GROUP

Cont: Paul Janson
P.O. Box 397, Dapto, 2630.

COMPUTERTOWN CAMDEN

Cont: Keith Stewart,
P.O. Box 47, Camden, 2570.

NEWCASTLE MICRO USERS GROUP

Cont: Dennis Jackson,
Tel: (049) 63 1910
Meet: Last Wed. monthly 7.30pm, Hall, Cnr. Fowler and Ogen Sts., Hamilton Sth.

PEACH USERS GROUP

Cont: 120 Lawson St. Redfern,N.S.W. 2016

SYDNEY PEACH USERS GROUP

Cont: 261 Northumberland St.
Liverpool, N.S.W. 2170.

VICTORIA

BALLARAT COMPUTER USERS GROUP

Cont: John Preston, Tel: (053)31 4363
Meet: 1st Wed. monthly at 7.30p.m.
Various venues - refer above.

EASTERN SUBURBS USERS GROUP

Cont: John Fletcher
Tel: Home (03) 737 9544
Bus (03) 89 0677 (9-4)
Meet: 4th Wed. monthly, 7.00pm. Kingswood College, 355 Station St. Box Hill.

GEELONG COMPUTER CLUB

Cont: P.O. Box 6, Geelong, 3220.
Meet: 2nd Thurs. monthly, Tybar Engineering Hampton St. Newton.

MICROCOMPUTER CLUB OF MELBOURNE

Cont: MICOM, P.O. Box 60, Canterburay, 3126
Meet: 3rd Sat. monthly, 2.00pm. Burwood State College, Burwood Hwy.

QUEENSLAND

TRS-80/SYSTEM 80 COMPUTER GROUP

Cont: Lance J. Lawes, 21 Rodney St. Lindum
Tel: (H)(07)396 2998 (W)(07)268 6811
Meet: 1st Sun. monthly 1:30pm at Lindum Progress Hall, Lindum Rd. Lindum,4178.

COMPUTER OWNERS GROUP

Cont: Betty Adcock, Tel: (07) 263 4268

TOWNSVILLE GROUP

Cont: Townsville Amateur Radio Club
Meet: 2nd Tues. monthly, State Energy Serv. HQ., Green Street, West End.

SOUTH AUSTRALIA

ADELAIDE MICRO USER GROUP

Cont: Rod Stevenson, 36 Sturt St. Adelaide.
Tel: 51 5241 between 9-4.

NORTHERN TERRITORY

DARWIN GROUP

Cont: Tony Domigan,
P.O. Box 39086, Winnellie, 5789.

NT 80 MICRO COMPUTER USERS GROUP

Cont: Mr. Harmon Venner, President,
96 Freshwater Rd. Jingili, Darwin, N.T.

WESTERN AUSTRALIA

CPU - THE CLUB FOR PEACH USERS

Cont: Brendon Butcher, Tel: (09) 367 5880

PERTH '80 USERS GROUP

Cont: C. Powell (09) 457 6849
Meet: 1st Tues. monthly, 7.30pm. Comm. Rec. Hall, MacDonald St. Yokine.

TASMANIA

DEVONPORT COMPUTER INTEREST GROUP

Cont: John Stevenson, Tel: (004) 92 3237

NORTH-WEST TASMANIAN USER GROUP

Cont: Rod McLeod, Tel: Home (004) 372064
Bus. (004) 301611

** UNITED KINGDOM **

COMPUTERTOWN NORTH-EAST

Cont: c/o 2 Claremont Pl. Gateshead, Co. Tyne & Wear NE8 1TL.
Tel: 0632-770036/643417/679119/559167.

COMPUTERTOWN UNITED KINGDOM

Cont: Dave Tebbutt, c/o 14 Rathbone Pl.
London W1P 1DE

INTERNATIONAL TRS-80 LEVEL I USER GROUP

Cont: Mr. N. Rushton, 123 Roughwood Dr.
Northwood, Kirley, Merseyside, L33 9U9.

NATIONAL TRS-80 USERS GROUP

Cont: Brian Pain, 40A High St. Stoney Stratford, Milton Keynes.

NEWCASTLE PERSONAL COMPUTING SOCIETY

Cont: John Stephen Bone - 0632 770036

NORTH-EAST TRS-80 USERS GROUP

Cont: Barry Dunn, 8 Ethick Tce. North
Craighead, Stanley, Co. Durham DH9 6BE.
Tel: 0207 30184.

NORTH WEST TRS-80 USERS GROUP

Cont: The Secretary,
40 Cowlees, West Houghton
Bolton, BL5 3EG.

TANDY OWNERS PROGRAM & INFORMATION CO-OP

Cont: Derek Higbee, 12 Shelley Close, Ashley Heath, Ringwood.
Tel: Ringwood 6720.

TRS-80 EDUCATIONAL USERS GROUP

Cont: Dave Futcher - Head Teacher,
Beaconsfield First & Middle School,
Beaconsfield Rd, Southall,
Middlesex.

WEST HERTS 80 USERS GROUP

Cont: Terry Bradbury, 20 Spruce Way,
St. Albans,
Herts.
Tel: Park St. 73663.

** NEW ZEALAND **

AUCKLAND

Cont: Ron Feasy, Bus. 799366 Home: 469455
Meet: 1st Tues. monthly, 7.30pm
N.Z. Solenoid Co. Ltd. 28 Kalmia St.
Ellerslie, Auckland.

WELLINGTON

WELLINGTON SYSTEM 80 USERS GROUP

Cont: Murray Trickett, Tel: 724-351 (W)
662-747 (H)
Meet: 2nd and last Tues. monthly.

- 0000000000 -

SOFTWARE

***** STAR TREK (Colour) by R.O. Edwards *****

This game of Star Trek has comprehensive instructions built into the game which need not be repeated here. The Colour Computer version has a couple of modifications in it that were necessary due to the small screen size. All of the computer responses to your actions will scroll across the screen in a small window; this response will only be replaced when you take any action that requires a new response. Also, the Galaxy Map, (being about three times the width of the normal screen), can only be viewed a portion at a time. Think of your screen as a window and using the arrow keys, move this window left and right to view the required portion of the Galaxy map page. One further note for the Colour Computer version: the program has a POKE that gives text on an orange background. If the program terminates by its intended means, the original value at the POKE address will be restored. If you should 'BREAK' the game, then it will be necessary to perform the following:

```
POKE 359,126
```

This will restore the green background; failure to do this could cause the computer to appear to lock up; if it does, you can still blindly type the above (though nothing will appear on the screen until you press ENTER). This will restore the screen to normal.

- 0000000000 -

***** THE ARISTOCRAT (Colour) by R. Dyball *****

The Aristocrat is a poker machine for the Colour Computer and the Hitachi Peach. In each version you start with a credit of $10; to start the game you press the space bar. To start a new game you press ENTER/RETURN. The Colour Computer version has coloured blocks instead of letter symbols; the colours correspond to the symbols in the following manner:

```
YELLOW = 10 (TENS)
BUFF   =  J (JACKS)
BLUE   =  Q (QUEENS)
RED    =  K (KINGS)
BLACK  =  A (ACES)
```

The payouts for each version are the same. The Colour Computer version has sounds for the reels turning, the handle being pulled down and the coins dropping into the tray. The Hitachi version has a simple tone that is sounded when the reels stop spinning. In each version play continues until you run out of money.

- 0000000000 -

***** URANIUM CORE - LII/16K by D.S. Stevens and B. Thomas *****

NOTE: This program is strictly for Level II.

Uranium core is set sometime in the future. Earth is rapidly running out of natural resources. Your mission is to retrieve the uranium cores found in the second universe and return them to our universe via the universe interface. Your mission will be deemed unsuccessful if no cores are present in the pod bay.

Your ship is the new 'Vectored Probability Shift Pod' and it uses V.P. shifts to move. To slow forward movement, you must select reverse. Likewise with up: select down to slow movement (and vice-versa).

You have only one life, so be careful! Watch out for unstable neutrino webs and black holes!

PROGRAM BREAKDOWN.

Lines 10-120	Program title and authors, system type input.
140-150	Print @ position conversion.
160	Move player.
490	Unsuccessful mission.
500	Successful mission. Calculate new fuel, reset velocity, increment level.
550	Explosion routine. Print "Game Over".
580	Read machine language data and create strings, define variables.
700	Start game, draw screen.
780	Calculate new core position, set black hole if necessary.
850	High scores.
880	Print high score congratulatory message, wait for player to input name or initials. Calculate player's position in high score array.
890	Print high score table.
1000	Print title page and await input (for instructions or game).
1230	Print instruction page and await input (to start game). (No input, so go to title page again).

The program uses SET (X,Y) co-ordinates and these are converted to give ?@ position. There are three machine language subroutines used by this program. The explosion routine is memory dependent so this routine may not be put elsewhere, but the sound and inverse screen routines are relocatable code, so you may move them if they conflict with any routines you like to have in your computer. Don't forget to rest the memory size!

1) EXPLOSION ROUTINE resides from 32305 to 32651

2) SOUND ROUTINE resides from 32688 to 32716

3) INVERSE ROUTINE resides from 32717 to 32767

STRING DATA

LB$ = Long Blank	WB$ = Web
UT$ = Uranium Core Title	P$ = Player's Ship
VB$ = Vertical Border	UC$ = Uranium Core
HB$ = Horizontal Border	BH$ = Black Hole
TB$ = Top Border	B$ = Blank

VARIABLE DATA

C1 = Arrow key location	SC = Score
C2 = and keys location	PX = Player's X position
C3 = Start of screen memory	PY = Player's Y position
C4 & C5 = USR entry points	PP = Print position (calculated from PX and PY)
I = Inverse constant for C4	XV = X Velocity
M = Sound constant for C4	YV = Y Velocity
EX = Explosion constant for C4	FP = Fuel Pointer
LV = Level	GF = Grapple Flag.

If you have any queries about this program, please don't hesitate to contact me. HAVE FUN!!

- 000000000 -

***** SINGLE KEY MENU - MODEL 3 DISK by S.J. Turtle *****

NOTE: This program is strictly for Model 3 Disk users and requires NEWDOS 80.

This is one for those people who get callouses on their fingertips from typing in all those DOS commands so necessary to get anything out of their machines.

When correctly set up, this program displays a list of up to 16 of your most commonly used programs and/or commands with a pointer which can be moved up or down to point to any one.

All the user needs to do is to move the pointer, using the up or down arrow key to the required command and press (ENTER) ... the command is then executed as if it had been typed in at NEWDOS 80 READY. Pressing the (CLEAR) key restores the usual NEWDOS 80 READY command mode, which is also the default after any DOS command has been executed.

The correct setting up of this routine requires that you are familiar with, and have available

an editor/assembler and SUPERZAP as some minor changes must be made to two system programs, namely SYS0/SYS and SYS1/SYS.

The first thing to do is get MENU into your machine, either by typing in the source code through an editor/assembler or by loading it from a MICRO-80 disk or cassette. I would recommend typing in the source code and saving it as it is necessary to use the editor/assembler to write your own commands into the menu ...more about that later. Save the source and assembled object code to disk ...it is necessary to have object module on the system disk.

***** BEFORE GOING ANY FURTHER make a backup copy of your system disk and use it to make the system program changes on, as any errors will render it useless until they can be repaired using Superzap. *****

Next enter SUPERZAP.....

1) Type DFS (ENTER)
2) Respond to FILESPEC? with SYS0/SYS
3) Respond to RELATIVE SECTOR #? with 8
4) Type MOD92 (ENTER)
5) Change 01 00 B2 4B to 01 FA B2 4B and save the change to disk
6) Move to the next sector...FRS 9
7) Type MOD8D (ENTER)
8) Change 00 00 00 00 00 00 01 00 B0 4C 00 to 01 05 00 F2 C3 40 00 01 FF B1 4C and save these changes to disk.
9) Press X to return to SUPERZAP menu.
10)Exit SUPERZAP and REBOOT THE DISK.

Now re-enter SUPERZAP and...

1) using the method in 1 to 3 above go to sector 1 of SYS1/SYS.
2) Type MOD2A (ENTER)
3) Change C3 40 00 to C3 00 F2 and save to disk.

You can now exit SUPERZAP...

If all is well you shouldn't notice anything! However, if you have made an error there are three most likely symptoms - first the screen will go blank except for NO SYS in the top left corner (you probably messed up number 5) or SYSTEM PROGRAM NOT FOUND repeatedly after a successful boot or constant reboots (you probably forgot number 8).

When NEWDOS 80 READY appears type MENU or LOAD MENU/CMD and press (ENTER) then when NEWDOS 80 READY returns press (ENTER) again and the menu should appear.

If you have made any mistakes with zapping the system programs, you probably won't have made it this far, so any faults from now on are most likely in MENU/CMD itself.

You will, of course, want to change your menu to suit yourself. This is best done using the editor/assembler - you will find in the source code all the commands in DEFM statements ORIGinated at F2F0H - by changing these to what you want and reassembling the program you can include any DOS command string that is permitted under NEWDOS 80, up to 40 characters long which is ample for most, if not all, needs.

To change the program it is only necessary to alter the MENU/CMD program. You never need to change the system programs once they have been set up, and the disk will still operate quite happily without the menu resident, provided you never overwrite the three byte jump at F200 ...for this reason I recommend using an AUTO chain routine that sets HIMEM to F1FEH and automatically loads MENU/CMD.

HOW IT WORKS

SYS1/SYS is the overlay module which is used to 'interrogate' the DOS commands and act upon them. It usually jumps to a ROM line input routine at 0040H from 4E24H. SYS1/SYS is only in the system when a DOS command is entered. In order, therefore, to permanently change this vector to the MENU routine, it is necessary to change SYS1/SYS itself - that is, the second of the two zaps - we alter the jump to 0040H (C3 40 00) to F200H, the entry point of MENU (C3 00 F2). Now we are left with a problem. If we try to boot the disk with that change only control will jump unconditionally to F200H every time, whether anything is there or not, hence constant reboots.

To overcome this situation we make a change to SYS0/SYS which is the module loaded first to set up the DOS vectors and the entire DOS environment. This change consists of placing three bytes at F200H to branch control back to 0040H until we get MENU Loaded - this is the first of the zaps.

A quick note here about disk file formats to enlighten the now totally confused novices - each block of code on a disk file contains a header of 4 bytes to tell the computer...

1) where to put it, and

2) how many bytes to put.

i.e. in our zap 01 05 00 F2 C3 40 00 means....

01 - this is a new section
05 - there are 5hex bytes including the two address bytes (the maximum is 00hex which equals 256)
00 F2 - where to put it...the address in LSB/MSB format, i.e. F200H
C3 40 00 - the code to be put, i.e. JP 0040H

The other four bytes 01 FF B1 4C were needed to load the next block, one byte of which we borrowed for our zap.

For anyone who would like to know more about disk files etc., I would recommend 'TRS-80 DISK AND OTHER MYSTERIES' by H.C. Pennington, which describes all types of disk files in great, yet easy to understand detail. (It is available from MICRO-80 for $27.00 and well worth every cent).

The only other zap to SYS0/SYS is to allow for the 6 bytes we borrowed from the last block.

The operation of MENU/CMD itself is fairly straight-forward and I won't discuss it in detail. The program listing documentation covers most of it. Basically the DOS routine at 4E24H jumps to F200H, the entry point for MENU which tests for a key input. If the key is anything other than (ENTER) it saves it and jumps to ROM at 05E3H to get the rest of the command and then continues as if MENU was not resident. If the key is (ENTER) it clears the screen, writes the menu and waits for up or down arrows, (ENTER) or (CLEAR). The arrows move the pointer in the appropriate direction, the (CLEAR) key clears the screen and returns control to the DOS routine by jumping to 0040H, the original branch in the unaltered DOS, whereas the (ENTER) key causes MENU to load the command to which the pointer is pointing into the input buffer at 4225H and then jumps back to DOS at 497BH, where it usually would after getting a command from the keyboard. DOS then continues as if the command had been typed in normally until it again arrives at 4E24H where the cycle starts again.

When in the menu mode it is possible to enter the usual command mode simply by pressing (CLEAR). After each command is completed the system is in this command mode. To enter menu mode simply press (ENTER) before any other key.

To change your menu, load the source code into your editor/assembler and alter the DEFM statements at the bottom of the listing. Always place a DEFB 0FFH after the last DEFB 00 byte to mark the last line of the menu, whether you have 16 entries or not. This is to stop the pointer at your last one and save it trying to point past the end of the screen.

- 0000000000 -

***** DEFUSR - LII/4K by Roger Bowler *****

NOTE: This program is strictly for Level II only.

This article describes an extension to Level II BASIC which should be of interest to anyone who makes USR calls from BASIC but particularly to those with a future move to Disk BASIC in mind.

If you ever upgrade your tape-based TRS-80 to a disk system, you will be pleased to find that Disk BASIC is generally "upward compatible" with Level II BASIC - that is to say, any program written according to the rules of Level II BASIC will normally run unchanged under Disk BASIC. This is very useful, because it means that you won't have to make any alterations to your existing programs to make them work on a disk-based TRS-80 system.

There is, however, one annoying exception to this rule, and it affects programs which call machine-code ("USR") subroutines. Level II BASIC and Disk BASIC use different methods for defining the entry address of a machine-code routine. Under Level II BASIC, you have to break the USR routine's address into its least-significant and most-significant bytes, and POKE these values into locations 16526 and 16527 (decimal) respectively, whereas Disk BASIC provides you with the somewhat more elegant DEFUSR statement.

The following example illustrates the two methods of defining the entry address of a USR routine starting at address 7D00 (32000 decimal).

Level II BASIC.....POKE 16526,0: POKE 16527,125

Disk BASIC.....DEFUSR=32000

What this means is that before you can run your Level II programs under Disk BASIC you have to go through them and convert all the POKE 16526 etc. lines into DEFUSR= statements. If you forget to do this, then not only will the program not work, it is quite likely that the POKE's will cause internal errors in the interpreter.

The ideal solution to this problem would be to provide Level II BASIC with the capability to handle the DEFUSR statement. Then you could write all your programs using DEFUSR, in the knowledge that they will work equally well under either BASIC. In fact, this turns out to be simpler than it sounds; it can be done by adding as little as 16 bytes of machine code to the Level II interpreter.

The first program listing shows the assembly source of the code necessary to interpret DEFUSR statements. (If you are not interested in how it works, you can skip the rest of this paragraph, which gives a brief outline of the routine's internal logic). The Level II interpreter takes a call to location 4158H whenever a statement beginning with DEF is encountered in a BASIC program. Normally this location contains a jump to a routine which issues the ?L3 ERROR message, but we shall overlay it with a jump into our routine instead. When the call is made, the HL register points to the location within the BASIC program which follows the word DEF. Our routine first checks that the next two words in the program are USR and = (?SN ERROR is issued if not), then it evaluates the expression following the =. The integer value of this expression is stored at locations 408E/FH (decimal 16526/7), thus defining the USR entry point. Finally, a return is made to the Level II interpreter with HL pointing to the end of the statement. Most of the processing is done by ROM calls which are more fully described in MICRO-80's LEVEL II ROM REFERENCE MANUAL. It is the power of these ROM calls which makes the DEFUSR routine so compact.

To implement DEFUSR, all you need to do is to poke the 16 byte program somewhere untouched by Level II (I suggest 4040-404FH which are used by Disk BASIC for storing the date and time), and place a jump instruction to it at 415BH. The BASIC program shown in the second program listing will do this for you. Lines 20 to 50 store the machine code in low RAM locations 4040-404FH and line 60 alters the last two bytes of the JP instruction at 415BH to jump to 4040H. Run this program once, and the computer will remain set up to accept programs containing DEFUSR statements (although you will have to rerun the program each time you power on the computer).

The third program listing is included as a demonstration to allow you to try out DEFUSR. All it consists of is the machine code from the 'white-out' program (given as a sample USR routine in section 8 of the TRS-80 Level II BASIC Manual) but the POKE 16526 etc. line has been replaced by a DEFUSR= statement. To use this program, power up the computer with a MEMORY SIZE of 31999, run program 2 to enable DEFUSR, then run program 3. The result is not particularly exciting but no doubt you will have plenty of programs of your own which will benefit from DEFUSR in a more spectacular way.

- 0000000000 -

***** COMPOUND MULTIPLICATION AND LONG DIVISION - LI/4K *****

by C. Stobert

This program was developed to generate problems for practice in long division and compound multiplication using conventional layout and procedure.

The introduction is sketchy as only 20 bytes of memory remain. No short cuts have been taken with punctuation so some extra memory may be found this way, though I prefer to "stay with the book".

The program selects numbers and presents them in calculation format and the answers are entered in turn as per conventional procedure.

Each sequence consists of ten separate problems and any treated incorrectly are repeated. The "mark" calculation is accurate for reasonably few errors as the counters calculate only the number of "turns" and "total correct".

Lines 5-120 Presents nominal introduction and selects type of problem to be processed.

200-270 Presents varying approvals if solution is correct.

280-350 Presents varying personal introductions to break the monotony (not as grand as L II with sound but LI-4K does have its limitations).

400-460 Selects the operating numbers for multiplication problems and breaks them into their component units, tens, etc. for placement in the operating format.

470-520 Places the numbers in the screen format and completes the detail with appropriate underlinings and operating sign.

560-640 Places the input prompt at the appropriate operating place in the format and then after each input replaces numbers removed by the scroll.

The program loops until all four input lines have been completed. The last input includes the units, tens, etc. components of the attempted result.

650-660 Accepts input of $x10^3$ and $x10^4$ components as appropriate.

670-680 Calculates result of input and compares with correct result.

690-720 Assigns program to problem repeat, new problem or conclusion as appropriate.

800-840 Conclusion sequence.

900-940 Data lines for location of format layout and 1st - 4th input lines respectively.

950-990 Subroutine to replace inputs removed by scrolling.

1300-1340 Subroutine for input of answer sequence for long division problems.

1400 Removes multiplication data locations to allow access to Data for long division operations.

1410-1465 Selects divisor and dividend and breaks them into component units, tens, etc.

1470-1590 Places working numbers in location on the operating format.

1600-1650 Completes format with appropriate separating lines.

1680 Inputs first portion of quotient (hundreds).

1690 Inputs first entry calculation and remainder.

1700 Transfers next number down from dividend.

1720-1780 Sequences input to final remainder calculation proceeding as above.

1800 Calculates final remainder.

1840-1850 Calculates answer and compares with the correct result.

1860-1890 Directs program to new problem etc.

2000-2060 Data for long division operating locations.

- 0000000000 -

***** STARSHOOT - L1/4K by M.S. Young *****

This program is an L1/4K version of the game as in MICRO-80 July, '81. The object of the game is to get a pattern of stars as shown below from the initial position, by shooting stars. Only stars can be shot. A star is shot by entering the number of its position. When a star is shot the pattern will change, depending on which star is shot.

Start	End	Layout
. . .	* * *	1 2 3
. * .	* . *	4 5 6
. . .	* * *	7 8 9

The above layouts are on the screen throughout the game for reference.

EXPLANATION OF PROGRAM

The board for the game is stored in an array A(1) - A(9). The value depends on if a star or a dot is in that position (1 or -1 respectively). A "second" array is used to store the print positions A(11) - A(19) for positions 1-9. The actual positions on the screen are READ from the DATA in line 9999.

Lines 1001 - 1009 make the changes for a shot in positions 1-9. The line is selected in line 580 with ON S GOSUB ....

Lines 2000 - 2050 print the board and determine the position of the game. Variable E is used for the status of the board.

If E = 7 (8 stars and 1 dot) AND the centre position A(5) is a dot, the game is completed (Line 620).

If E = 9 (9 dots no stars). This is also an end of the game, as no stars are left to shoot (Line 600).

Lines 3000 - 3160 print the instructions.

Lines 4000-4060 print the layouts.

The game can be completed in 11 moves, but bad shooting may clear the board and end the game. Good Luck!

The line numbers may look rather odd, but there is a reason for the numbers used.

The main program starts at line 500 and other sections start on lines 1000, 2000, 3000 and 4000.

I find it easier to follow the logic used in a program if the lines are numbered in this way rather than starting at line 10 with an increment of 10.

- 000000000 -

```
      **** STAR TREK ****

        COLOUR COMPUTER

10 ' **     STAR TREK     **
20 ' ** R.O. EDWARDS      **
30 ' ** 31 REDESDALE RD  **
40 ' ** IVANHOE     3079 **
50 'MODIFIED FOR THE COLOR
    COMPUTER BY MICRO-80
60 CLEAR3000:RESTORE:CLS:PRINT@2
01,"STAR TREK":GOSUB1380:FORI=1T
01000:NEXT
70 MX=1:BL$=STRING$(20," "):SD$=
BL$+"* * * * * * * * * * *"+BL$
80 CLS:PRINT@192,"DO YOU WANT IN
STRUCTIONS, (Y,N)?":GOSUB1030:IF
Q$="Y"THEN1110
90 PRINT@224,"NO OF KLINGONS,  1
(LOW)-5(HIGH)";:GOSUB980:N=VAL(I
N$):IFN<1ORN>5THEN90ELSE CLS:PRI
NT@229,"CREATING GALAXY":Q=1.8IN
100 K9=0:B1=0:K1=0:T0=(RND(20)+2
0)*100:T=T0:FORI=1TO8:FORJ=1TO8:
R= RND(500)/Q
110 K=0:IFR<20THEN K=1:K1=K1+1:I
FR<10THEN K=2:IFR<5 THENK=3
120 K9=K9+K:B=0:IF RND(0)>1-N/50
 THENB=10:B1=1
130 G(I,J)=K*100+B+RND(9):Z$(I,J
)="   ":NEXTJ,I:K0=K9:IFK1>600RK
1<3THEN100
140 Q1=RND(8):Q2=RND(8):IF INT(G
(Q1,Q2)/100)>0THEN140ELSE IFB1<>
1 THENI=RND(8):J=RND(8):G(I,J)=G
(I,J)+10
150 CLS:T9=K9*3:S1=RND(8):S2=RND
(8)
160 PRINT@96,"PRESENT STARDATE"T
:PRINT"AS COMMANDER OF THE U.S.S
. ENTE-RPRISE YOUR MISSION IS TO
 RID   THE GALAXY OF THE DEADLY
KLINGONMENACE, TO DO THIS, YOU M
UST    DESTROY THE KLINGON INVAS
ION    FORCE OF"K9"BATTLE CRUISE
RS."
170 PRINT"YOU HAVE"T9"SOLAR YEAR
S":PRINT"TO COMPLETE YOUR MISSIO
N        (I.E. UNTIL STARDATE"T0
+T9")":GOSUB970
180 FORI=0TO5:READD$(I):D(I)=5:N
EXT:E=3500:D=500:P=10:P$= CHR$(9
5)+"      ":I$= STRING$(21,45):DA
TAWARP ENGINES,SECTOR SCANNER,DE
EP SPACE SCANNER,PHASER CONTROL,
PHOTON TUBES,SHIELD CONTROL
190 X=G(Q1,Q2)/100:K= INT(X):B=
INT((X-K)*10):S=G(Q1,Q2)- INT(G(
Q1,Q2)/10)*10
200 FORI=1TO8:FORJ=1TO8:Q(I,J)=0
:NEXTJ,I:FORI=1TO3:K(I,3)=0:NEXT
210 Q(S1,S2)=1:IFK>0 THEN FORI=1
TOK:GOSUB940:Q(R1,R2)=2:K(I,1)=R
1:K(I,2)=R2:K(I,3)=200:NEXTI
220 IFB>0THENGOSUB940:Q(R1,R2)=3
230 GOSUB860:IFS>0THEN FORI=1TOS
:GOSUB940:Q(R1,R2)=4:NEXT
240 IFD(2)>0 THEN FORI=Q1-1TOQ1+
1:FORJ=Q2-1TOQ2+1:Z$(I,J)=RIGHT$
(STR$(G(I,J)),3):NEXTJ,I
250 CLS:PRINT@0,"SHORT RANGE SCA
N":FORI=1TO8:PRINT@I*32,"";:IFD(
1)>0 THEN FORJ=1TO8:PRINT MID$("
.EAO*",Q(I,J)+1,1);:NEXTJ
260 ON I GOSUB290,300,310,320,33
0,340,350,360:NEXTI
270 PRINTI$;:IFD(1)<=0 THEN SD$=
BL$+"SHORT RANGE SCAN INOPERABLE
"+BL$
280 GOTO390
290 PRINT"STARDATE ";T:SP=416:Q=
Q1-1:GOSUB370:RETURN
300 PRINT"CONDITION "C$:RETURN
310 PRINT USING"QUADRANT  #:#";Q
1,Q2:Q=Q1:SP=448:GOSUB370:RETURN
320 PRINT USING"SECTOR    #:#";S
1,S2:RETURN
330 PRINT"ENERGY    "E;:Q=Q1+1:SP
=480:GOSUB370:RETURN
340 PRINT"TORPEDOES"P:RETURN
350 PRINT"SHIELDS   "D:RETURN
360 PRINT"KLINGONS "K9:RETURN
370 IFD(2)>0 THEN PRINT@SP, USIN
G"##############";G(Q,Q2-1),G(Q,Q
2),G(Q,Q2+1);ELSE IFQ=Q1 THEN SD
$=BL$+"LONG RANGE SCAN INOPERABL
E"+BL$ ELSE PRINT
380 RETURN
390 MX=1:PRINT@384,"LONG RANGE S
CAN";:PRINT@328,"          ";: PRI
NT@320,"COMMAND?";:GOSUB410:Z$(Q
1,Q2)=RIGHT$(" "+ STR$(G(Q1,Q2))
,3)
400 Q$=INKEY$:GOSUB1450:IFQ$=""T
HEN400ELSE A=ASC(Q$)-48:IFA<0 OR
 A>5 THEN400ELSE IFA<4 AND A>0 A
ND D(A+2)<=0 THEN SD$=BL$+"**"+D
$(A+2)+" INOPERABLE **"+BL$:GOTO
390ELSE ON A+1GOTO430,570,630,77
0,790,800
410 PRINT@311,"COMMANDS:";:PRINT
@342,"0 COURSE ";:PRINT@374,"1 P
HASERS ";:PRINT@406,"2 PHOTON  "
;:PRINT@438,"3 SHIELDS ";:PRINT@
470,"4 DAMAGE ";:PRINT@502,"5 MA
P ";:RETURN
420 PRINT@310,"     3     ";:PRIN
T@339,"     4  :  2";:PRINT@371,
"       *:*  ";:PRINT@403,"    5
---*---1";:PRINT@435,"       *:*
  ";:PRINT@467,"     6  :  8 ";:
PRINT@499,"        7   ";:RETURN
430 GOSUB420:MX=1:SD$=BL$+"SET C
OURSE (1-9)"+BL$:PL=334:GOSUB990
:C=QQ:IFC<1THEN390ELSE IFC>9THEN
430
440 SD$=BL$+"WARP FACTOR (.1 - 1
2)"+BL$:PL=332:GOSUB990:W=QQ:IFW
<=0THEN390ELSE IFW>12THEN440
450 IFW>1 AND D(0)<=0THEN SD$=BL
$+"WARP ENGINES DAMAGED, MAX SPE
ED WARP 1"+BL$:FORI=1TO1000:NEXT
:GOTO390
460 CLS:T=T+1:IF T>T0+T9 THEN104
0ELSE FORI=0TO5:D(I)=D(I)+.5:IF
D(I)>5THEN D(I)=5ELSE IF D(I)<-5
THEND(I)=D(I)+.5
470 NEXT:IF RND(50)<W THEN D=D-R
ND(1100)-200:SD$=BL$+"*** SPACE
STORM *** "+BL$:IFD>0 THEN SD$=B
L$+"*** SHIELDS HELD ***"+BL$ EL
SER=RND(6)-1:D(R)=D(R)+D/100:SD$
=BL$+D$(R)+" DAMAGED ***"+BL$:D=
0
480 N= INT(W*8):E= INT(E-N*2):IF
E<1THEN1040ELSEQ(S1,S2)=0:X=S1:Y
=S2:GOSUB760:FORI=1TON:S1=S1+X1:
S2=S2+X2
490 IFS1<.5ORS2<.5ORS1> =8.5ORS2
> =8.5THEN530
500 IFQ(INT(S1+.5),INT(S2+.5))>1
THEN SX$=STR$(S1):SY$=STR$(S2):S
D$=BL$+"ENTERPRISE IS BLOCKED BY
 OBJECT AT SECTOR "+SX$+":"+SY$+
BL$:S1=INT(S1-X1+.5):S2=INT(S2-X
2+.5):FORI=1TO1000:NEXT:GOTO520
510 NEXTI:S1= INT(S1+.5):S2= INT
(S2+.5)
520 Q(S1,S2)=1:GOSUB860:GOTO240
530 X0=Q1+W*X1:Q1=FIX(X0):Y0=Q2+
W*X2:Q2=FIX(Y0):S1=FIX(X+(X0-Q1)
*8+.5):IFS1<1 THENS1=S1+8:Q1=Q1-
1ELSE IFS1>8 THEN S1=S1-8:Q1=Q1+
1
```

MICRO-80 PRODUCTS CATALOGUE

This catalogue contains a selection from the wide range of peripherals, interfaces, computers and software carried by MICRO-80 for your computer. If you don't see the item you want, contact us, we probably have it anyway!

MICRO-80 has been supplying customers throughout Australia and the Pacific region by mail-order for 2½ years. Our customers find this a simple and efficient way to do business. You may place your order by telephone or by mailing the order form from any issue of MICRO-80 magazine. Generally, it takes about one week from receipt of order until despatch. You should allow 2-3 days for your letter to reach us and 7-10 days for the parcel to reach you, making a total turnaround time of 2½-3 weeks.

WARRANTY AND SERVICE

All hardware products carry a 90 day parts and labour warranty either from the manufacturer/distributor or from MICRO-80 Pty Ltd. In many cases, warranty servicing can be arranged in your own city, otherwise goods will be repaired by our own team of technicians in our Adelaide workshops.

TRADE-INS AND TERMS

MICRO-80 can accept your existing equipment as a trade-in on new equipment. We can also arrange consumer mortgage financing or leasing on larger hardware purchases. Contact us for details.

LNW SYSTEM EXPANSION II

for the Model 1.

The LNW Expansion for the Model 1 comes fuily assembled in an attractive, heavy gauge steel case, complete with power supply and full documentation. High quality construction throughout, the Expansion II features a full 32K of 200nS RAM, RS-232-C and printer interfaces, gold-plated connectors and a floppy disk controller.

PRICED AT $550 plus $10 freight

PRODUCTS FOR THE LNW80 COMPUTERS

SOFTWARE:

CHARM **$55.00 plus $2.00 p&p**
A programmable character generator for designing character sets, symbols and graphic characters with a maximum of ease and flexibility.

AUTOPLOT **$125.00 plus $2.00 p&p**
Autoplot enables you to make use of the high resolution capability of the LNW 80 more easily, with the ability to produce hard copies on a suitable printer with bit mapped graphics.

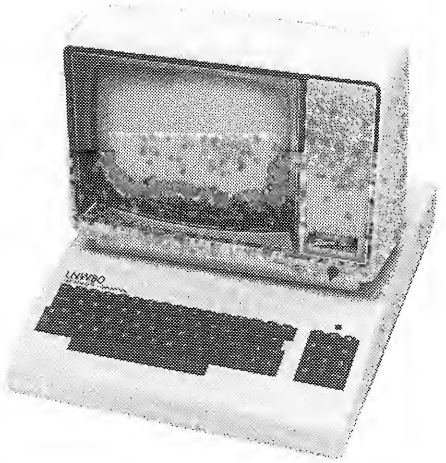
AVAILABLE SOON:

LNW Research Corp. has announced the release of several new products for their range of LNW80 Computers:

- CP/M Upgrade Kit
  To upgrade LNW80 "kit computers" to 96K of RAM and provide CP/M operation. This kit consists of a CP/M adaptor board, 64K of RAM, installation instructions, CP/M2.2 diskette and LNW CP/M manual.
- LNW LDOS
  A new LDOS operating system for the LNW80 Model 2 providing software compatibility with the TRS-80 Model 3. Users with the new CP/M upgrade kit will also be able to run the new LDOS.
- CP/M Plus
  CP/M Plus (or CP/M 3.0) is the latest version of the CP/M operating system and includes a CP/M Plus diskette and manual.

All prices include Sales Tax and are subject to change without notice. Some items are in limited supply. All equipment carries MICRO-80's Australia-wide 90 day warranty covering parts and labour.

Introducing the NEW LNW80 MODEL 2



The new LNW Model 2 is not just a microcomputer but a complete computer package that includes excellent hardware, extensive systems software and a range of application software. Manufactured by LNW Research Corporation, the LNW Model 2 features:

HARDWARE:

- 4MHz Z80A microprocessor with 96K user RAM.
- 16K x 6 bits Graphics RAM, expandable to 64K (four pages).
- Printer, RS-232-C and cassette interfaces.
- 12K BASIC in ROM (Level 2 compatible).
- Support for single/double sided, single/double density, 5¼" or 8" disk drives.
- Full TRS-80 Model 1 compatibility.
- Hi-res Colour (RGB) and B & W video outputs.
- Four Hi-Res Graphics Modes:
  — B & W 128 x 48
  — B & W 480 x 192
  — 8 colour 128 x 192
  — 8 colour 480 x 192
- Three text displays Modes:
  — 64 char x 16 lines
  — 80 char x 16 lines
  — 80 char x 24 lines

APPLICATION SOFTWARE:

1. LNW Small Business and Professional Accounting Series — *including General Ledger, Accounts Receivable, Accounts Payable and Payroll (U.S.A. conventions).*
2. Electric Spreadsheet — *for financial planning and forecasting.*

SYSTEMS SOFTWARE:

LNWBASIC 4.0 — an extension to disk BASIC that allows full use of the LNW80's graphics capabilities through the addition of powerful graphics commands such as CIRCLE, COLOR, DRAW, etc.

DOSPLUS 3.4 — the fast, reliable and easy-to-use operating system that provides all the file control and disk management you need for maximum benefit from your disk drives as well as an enhanced disk BASIC.

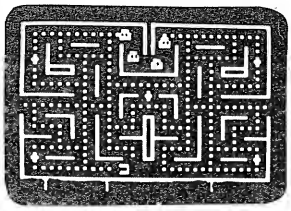
LNW-CP/M — the CP/M operating system opens the door to a whole new world of software. LNW-CP/M was designed to be compatible with application programs written for CP/M 2.2 and provides the user with a 61K system

3. Electric Pencil — *an easy-to-use word processor.*
4. Microterm — *an intelligent terminal program for communications.*
5. Chart-Ex — *allows you to transform your data into pie, line or bar charts on hi-res display or printer with bit graphics.*

The LNW80 Model 2 is perfect for the serious hobbyist or businessman seeking a higher performance, more reliable computer to replace his TRS-80 Model 1 without sacrificing a huge investment in software and programming experience.

LNW80 Model 2 computer *(complete except for disk drives and monitor)*	$2,850
HI-RES Green Phosphor Monitor	$265
Super HI-RES Hitachi RGB Monitor	$1,250
Two Single-sided 40 Track Double Density Disk Drives *(in cabinet with power supply and cable)*	$825

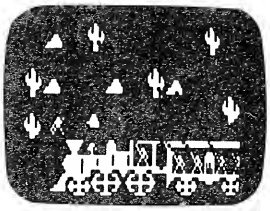
All prices include Sales Tax and are subject to change without notice. Prices are FOB Adelaide. Add $20 road freight anywhere in Australia. All equipment carries MICRO-80's Australia-wide 90 day warranty covering parts and labour.



SCARFMAN

This incredibly popular game craze now runs on your TRS-80! It's eat or be eaten. You run Scrarfman around the maze, gobbling up everything in your path. Try to eat it all before nasty monsters devour you. Excellent high speed machine language action game from the Cornsoft Group. With sound.

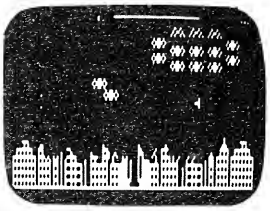
Price: $17.95



THE WILD WEST

It's up to you to keep the West beautiful with Outlaws and renegade Indians on all sides. Even the train has been captured by Outlaws with all the payroll on board. Can you clean up the Wild West?

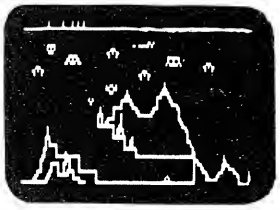
Price: $26.50



SPACE ATTACK

Steady your nerves, keep a sharp lookout, and prepare for battle to save your city. Fiendish aliens are all around, and if they destroy the city you lose.

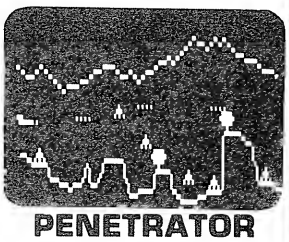
Price: $26.50



STRIKE FORCE

As the primary defender of a world of cities under deadly alien attack, your weaponry is the latest rapid fire missiles, long range radar, and incendiary "star shells." Your force field can absorb only a limited number of impacts. A complex game of strategy, skill and reflexes from Melbourne House.

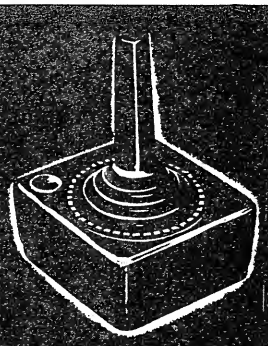
Price: $26.50



PENETRATOR

Soar swiftly over jagged landscape, swooping high and low to avoid obstacles and enemy missiles attacks. With miles of wild terrain and tunnels to penetrate, you're well armed with bombs and multiple forward missile capability. From Melbourne House. Features sound, trainer mode and customizing program.

Price: $36.50



STICKEROO JOYSTICK INTERFACE

for the TRS-80 MODEL I and SYSTEM 80

FROM **$32.00**

ADD $2.00 p. & p.

CONVERT YOUR COMPUTER INTO AN ARCADE GAMES MACHINE

Micro-80's Stickeroo Interface Features:

•Compatible with Joysticks for Atari, Vic-20 and most video games •Saves your keyboard from abuse •Compatible with programs from leading US software houses: Big Five, Cornsoft, Melbourne House, Adventure International •Adds a whole new dimension of pleasure and fun to your favourite games •Will be supported in MICRO-80 •Can be used with your own basic or ML Programs •Comes complete, ready to plug in and use •Absolutely no modifications required to your computer

Due to popular demand, Stickeroo Interface is now available separately so you can use the Joystick of your choice.

PRICE INCLUDES ... STICKEROO + INSTRUCTIONS + DEMO PROGRAM LISTING

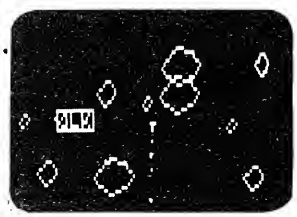
PLEASE SPECIFY TRS-80 MODEL I OR SYSTEM 80 WHEN ORDERING

The Stickeroo Interface plugs in to the expansion edge connector and may not be suitable for expanded systems.

PISTOL GRIP JOYSTICK WITH FIRE BUTTON

$25 + $2 p & p (No p & p required if ordered with Stickeroo Interface)

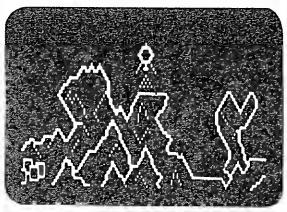
ALL GAMES ADVERTISED ON THIS PAGE ARE STICKEROO COMPATIBLE



SUPER NOVA

Asteroids float ominously around the screen. You must destroy the asteroids before they destroy you! (Big asteroids break into little ones). Your ship will respond to thrust, rotate, hyperspace and fire. Watch out for that saucer with the laser! As reviewed in May 1981 Byte Magazine.

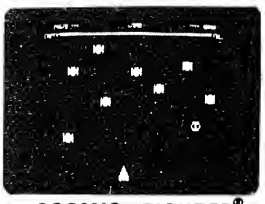
Price: $26.50



LUNAR LANDER

As a vast panoramic moonscape scrolls by, select one of many landing sights. The more perilous the spot, the more points scored -- if you land safely. You control LEM main engines and side thrusters. One of the best uses of TRS-80 graphics we have ever seen. From Adventure International. With sound.

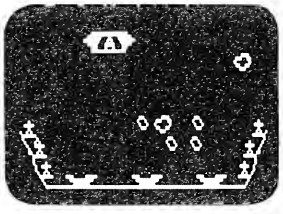
Price: $26.50



COSMIC FIGHTER®

Your ship comes out of hyperspace under a convoy of aliens. You destroy every one. But another set appears. These seem more intelligent. You eliminate them too. Your fuel supply is diminishing. You must destroy two more sets before you can dock. The space station is now on your scanner. With sound!

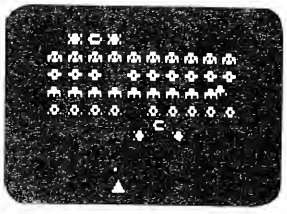
Price: $26.50



METEOR MISSION II

As you look down on your view, astronauts cry out for rescue. You must maneuver through the asteroids and meteors. (Can you get back to the space station?) Fire lasers to destroy the asteroids, but watch out, there could be an alien Flagship lurking. Includes sound effects!

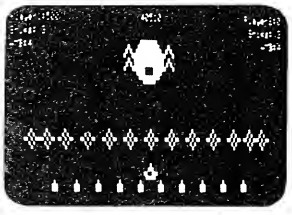
Price: $26.50



GALAXY INVASION

The sound of the klaxon is calling you! Invaders have been spotted warping toward Earth. You shift right and left as you fire your lasers. A few break formation and fly straight at you! You place your finger on the fire button knowing that this shot must connect! With sound effects!

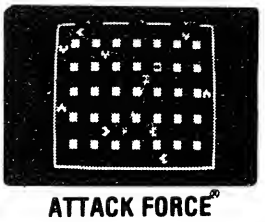
Price: $26.50



DEFENSE COMMAND

The invaders are back! Alone, you defend the all important nuclear fuel canisters from the repeated attacks of thieving aliens, repeatedly. An alien passes your guard, snatches a canister and flys straight off. Quick! You have one last chance to blast him from the sky! With sound and voice.

Price: $26.50



ATTACK FORCE®

As your ship appears on the bottom of the maze, eight alien ships appear on the top, all traveling directly at you! You move toward them and fire missiles. But the more aliens you destroy, the faster the remaining ones become. If you get too good you must endure the "Flagship". With sound effects!

Price: $26.50

FOR YOUR ENTERTAINMENT

MICRO-80 now offers you the widest range possible in entertainment software. These programs are supplied on cassette for the Level II/16K TRS-80 Model I/III (except as noted). They are also suitable for the System 80 but sound may not be available unless a hardware modification has been fitted to reverse the roles of recorders #1 and #2. *Order yours now while stocks last!*

DEFENCE PENETRATOR $20.95
DEFENCE PENETRATOR is based on one of the most popular arcade favourites of all time with smooth graphics and sound effects. With realistic scrolling planetscape it's the best game yet.

DEVIL'S TOWER $25.95
Aliens move in waves of 5 attackers with their robot scouts attacking you from the mountain, their war machines and their protector ships putting up force fields to protect them. Only your skill and fast reflexes can save the plant.

BATLE STATION $21.50
The aim of the game is to defend your space station against the attack of four alien space ships.

MORGOTH $20.95
Morgoth is a unique action packed adventure allowing you to wander through the enchanted dominion of Morgoth and collect the lost treasures of KAZARD KALLAHAN. But Beware! You must escape before the satanic Morgoth is aroused and seeks yea!

KILLER BEETLES $21.50
The aim of the game is to dig traps. When a beetle falls in you must fill it in to bury them, before they can catch you.

STAR CRESTA $20.95
Star Cresta takes you beyond the limits of your computer and into the Cosmic void itself! Beware! Iron clad concentration and lightning relfexes are required to destroy the evil empress.

JUNGLE RAIDERS $21.50
The aim of the game is to defend your four bases from the marauding Jungle Raiders. Your skill all the Jungle Raiders and they try to hit you with their spears or drag off all four of your bases.

ALIEN TAXI $28.50
Your goal is to pick up and deliver passengers to an underground resort hotel. There is a fare at each of the 12 taxi stands on the first level and 12 more on the second level.

KILLER GORILLA $21.50
Four completely different frames. Each one offering a different challenge, makes this one of the most complex and stimulating games ever written for a TRS-80. The game keeps track of the top ten scores along with a six character name for each score.

JUNGLE BOY $21.50
The ultimate challenge! Are your reflexes fast enough to swing Jungle Boy from vine to vine? Can you swing through the jungle? Can you swim by the alligators? These are just some of the things you will find very challenging in Jungle Boy.

STELLAR WARP $20.95
Animation with superior fighter craft brings you an even greater challenge. As your computer advances your level, the aliens become more dangerous and the harder it is to stay alive!

HOPPY $21.50
The aim of the game is to get your frogs across the busy highway without being squashed and then across the river by means of floating logs and turtles.

PANIK $28.50
Your mission is to rid the galaxy of the Mzors forever. Mzors are half animal and half machine. Their leaders are very difficult to destroy and are capable of creating more warriors at will. Your weapons are your energy pistol, short range transporter pack and your courage.

INSECT FRENZY $21.50
The aim is to stop the centipede from getting you, all the time keeping an eye out for the giant spider.

ALIEN CRESTA $21.50
The aim is to defend your ship from numerous attacks from an assortment of aliens. If you get hit three times, it's all over.

DESERT PERIL $28.50
The Zagons have mined the desert and have put killer satellites, drone bomber balloons, and flying dragons along the whole trail. The future of your planet's race depends on your skill and daring.

RALLY RACER $20.95
Drive through an action packed maze and try to hit all the flags before Morgan the Mad motorist or Crazy Harry and his killer hoodlums catch you!

NOTE:
As the prices of imported software may vary, these prices are valid for current stock only and prices are subject to change without notice.

Double Your Disk Storage Capacity with the

LNDoubler 5/8

The LNDoubler is easily installed into your expansion interface and provides support for both 5¼" and 8" disk drives. Completely compatible with all the major Disk Operating Systems, the LNDoubler provides technically advanced, tested and reliable double-density operation with such features as:

- Analog phase lock loop data separation.
- Precision write precompensation.
- High quality PCB will all contacts gold-plated.
- Drives 1-3 may be software selected as 5" or 8" drives and a switch is provided for drive 0.
- Supports any mix of 5" or 8" drives, single or double density, single or double sided.

The LNDoubler will inrease the formatted storage capacity of **each** 40 track single-sided drive by 80% to over 180Kbytes — for just over half the cost of one disk drive. With an 8 inch double-sided double-density disk drive, you can have over one Megabyte of online storage!

The LNDoubler 5/8 doubler with documentation is available for ...

$285 plus $2.00 p.&p.

NOTE: A special cable is required for 8" drive operation and 8" double-density operation requires a 3.55 MHz CPU speed-up modification.

now available ...

DOSPLUS 3.5

is the state of the art in Disk Operating Systems for the Model 1 and Model 3 offering an order of magnitude increase in flexibility and performance over its predecessor DOSPLUS 3.4 and yet, is easier to use and more friendly with a Help facility explaining the syntax of DOS commands. The huge manual of over 350 pages describes the system in detail and is sectioned and tagged so that you can find what you want more quickly. Far greater flexibility is offered by the introduction of device drivers that are external to the system and that can be tailored to your needs.

Some of the features offered by DOSPLUS 3.5 are ...

- Single and double density support with density recognition.
- Improved file control facilities and date stamping of files.
- A keyboard driver that offers single key entry.
- An extensive Job Control Language.
- Complete and detailed technical system information.
- Two versions of BASIC, plus a BASIC label facility.
- and much more.

DOSPLUS 3.5 REPRESENTS EXCELLENT VALUE AT $160

When ordering by MAIL please specify Model 1 or Model 3 and include $2 for freight

Run your own bulletin board with ...

MICRO-80 BULLETIN BOARD SYSTEM $119.00 + $2.00 p&p

MICRO-80 (!) is an advanced Bulletin Board System that can be configured to be public allowing anyone to use the system, to be private allowing only authorised access or to allow limited access to some users. As well as providing facilities for uploading/downloading program/data files and messages, MICRO-80 features multi-level system security, user passwords, provision for private messages and permanent user records, and automatically reclaims space when messages are deleted. MICRO-80 is provided with a minimum DOSPLUS operating system kernel.

MICROTERM 1.4 $119.00 + $2.00 p&p

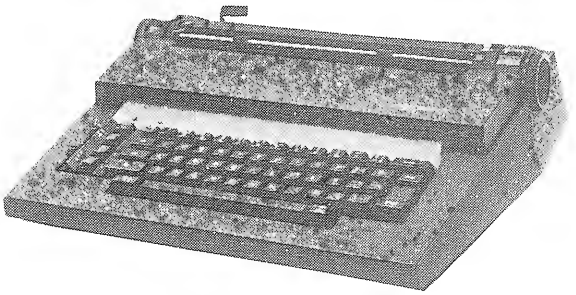
The intelligent microcomputer terminal package that provides many unique features such as Auto Dial and Automatic Transmission Time to send files to a remote site at any time, automatically. Microterm is provided with a minimum DOSPLUS operating system kernel and utility programs to upload and download both program and data files, and allows you to execute DOS commands without exiting the program. The input buffer continues to collect incoming data while going from terminal mode to command mode. The Model 3 version is certified to operate at speeds up to 4800 baud and the Model 1 version up to 600 baud (with no nulls inserted).

(Minimum System Requirements: 48K Model 3, 2 Drives, RS-232-C and Modem)

All prices include Sales Tax and are subject to change without notice. Some items are in limited supply. All equipment carries MICRO-80's Australia-wide warranty covering parts and labour.

Daisy Wheel Printers/Typewriters

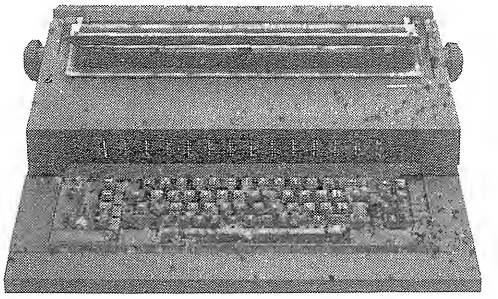
OLIVETTI PRAXIS 35



$895

plus $10 road freight anywhere in Australia

OLIVETTI ET-121



$1500

plus $20 road freight anywhere in Australia

MICRO-80 has converted these OLIVETTI typewriters to work with the TRS-80, SYSTEM 80 or any other microcomputer with a Centronics parallel port. Now you can have the best of both worlds — an attractive, modern, correcting electronic typewriter which doubles as a correspondence quality Daisy Wheel printer when used with your microcomputer.

The **PRAXIS** is a portable typewriter, designed for private and light commercial use with an average print speed of 6.5 c.p.s.

The **ET-121** is a large typewriter intended for heavier duty and features a print speed of up to 11.5 c.p.s.

Centronics printer cable to suit TRS-80 or SYSTEM 80 . . . . . $39

MICRO-80 is an A-Grade Olivetti distributor and has been producing printer conversions for Olivetti daisy wheel typewriters for several years. Write or call for full details.

16K Memory Upgrade Kit

$30

plus $2.00 p. & p.

Large volume means we can buy better and can pass the savings on to you. There are our proven, prime, branded 200 nanosecond chips, guaranteed for 12 months. A pair of DIP shunts is also required to upgrade CPU memory in the TRS-80 — these cost an additional $4.00. All kits come complete with full, step-by-step instructions which include labelled photographs. No soldering is required. You do not have to be an experienced electronic technician to install them.

Lower Case Modification

$49

plus $2.00 p. & p.

The MICRO-80 modification features true below-the-line descenders, a block cursor and symbols for the 4 playing-card suits. Each kit comes with comprehensive fitting instructions and two universal lower-case driver routines on cassette to enable you to display lower case. These routines are self-relocating, self-protecting and will co-reside with other machine language programs (the second includes keyboard-debounce and flashing cursor). Fitting requires soldering inside the computer and should only be carried out by an experienced hobbyist or technician. A fitting service is available in capital cities for only $20.00 and a list of installers is included with each kit. (Specify TRS-80 Model I or System 80 when ordering.)

All prices include Sales Tax and are correct at time of publication but are subject to changes without notice. All equipment carries MICRO-80's Australia-wide 90-day warranty covering parts and labour.

DISK OPERATING SYSTEMS & DEVELOPMENT SOFTWARE

You can increase your programming productivity, the execution speed and 'user friendliness' of your programs by using an enhanced Disk Operating System (DOS). Together with the other utility software, you can get the most from your disk drives.

DOSPLUS 3.4 **$149.95**
(Specify Model I single/double density or Model III)

A powerful DOS that provides many features and comes with a stand alone manual. With a high-degree of compatibility with TRSDOS, DOSPLUS 3.4 is suitable for the first-time or experienced user.

DOSPLUS 3.5 **$160.00**
(Specify Model I or Model III)

DOSPLUS 3.5 is a powerful, sophisticated DOS intended for the experienced user. The system can be configured to suit your requirements, provides greatly enhanced features over 3.4 and new features like single-key entry, date-stamping of files, a Help file and more. More user friendly than 3.4, DOSPLUS 3.5 comes with a very extensive stand-alone manual.

ENHBAS **$52.95**

ENHBAS adds over 30 new commands and functions to your BASIC interpreter including high speed SORT, labels in BASIC, RESTORE to any line number, WHILE-WEND for structured programming, SCROLL, LEFT, INVERT, DRAW and PLOT to give you ease of control over graphics, SOUND and PLAY to add realistic sound effects and many more. Makes programming a breeze! Available for Model I or III, disk or cassette — specify which when ordering.

NEWDOS 80 VERSION 2.0 **$185.00**
(Specify Model I or Model III)

Newdos 80 suits the experienced user who has already used TRSDOS, understands the manual and is prepared to learn the somewhat complicated syntax of one of the most powerful DOS's available. With the correct hardware, Newdos 80 supports any mix of single- or double-sided, single or double density, 5" or 8" disk drives with track counts up to 96. It provides powerful, flexible file handling in BASIC including variable length records up to 4096 bytes. Definitely not for the beginner.

MASTER DISK DIRECTORY **$20.95**

FIND THE PROGRAM FAST!! PAYS FOR ITSELF BY RELEASING REDUDANT DISK SPACE!! MASTER DIRECTORY records the directories of all your individual disks onto one directory disk. Then it allows you examine them, find an individual file quickly, list files alphabetically, weed out redundant files, identify disks with free space, list files by extension, etc., etc. This program is invaluable for the serious disk user and will pay for itself many times over. Not fully compatible with NEWDOS 80.

THE FLOPPY DOCTOR/MEMORY DIAGNOSTIC
Model III Disk **$43.50**

THE MICRO CLINIC offers two programs designed to thoroughly check out the two most trouble-prone sections of the TRS-80 — the disk system (controller and drives) and the memory arrays. Both programs are written in Z80 machine code and are supplied together on diskette for a minimum 32K, one disk system.

Note: For DOSes, include $2.00 for freight.

MORE ENTERTAINMENT SOFTWARE

BOSKONE ALERT **$25.50**

You have total control of every aspect of your fighter and must use your laser to destroy 9 Deathstars before the Earth comes into range.

OUTLAND **$25.95**

You must use your skills, reflexes and an array of weapons to defend your colony against the attacks of Xenos Star Raiders and prevent its destruction.

STELLAR WARP **$20.95**

Use your fighter craft to destroy the aliens who become more dangerous as your level advances. Beware of the space mines. In an emergency, activate Stellar Warp.

DOOMSDAY MISSION **$25.50**

You must disarm a number of nuclear missiles left by saboteurs on one of our space stations. Any direct assault on the station could launch those missiles.

CP-80 DOT MATRIX PRINTER

Features:
- 80 cps bi-directional, logic seeking
- 40, 71, 80 or 142 characters per line
- Normal and italic alphanumeric, symbol and semi-graphic characters
- Unidirectional bit image graphics (8 x 640 or 8 x 1280 dot/line)
- Tractor and friction feed

EPSON MX80 compatible control codes

FOR A LOW $599

SPECIAL ANNOUNCEMENT

We are expanding our range of disk drives to include the new

SLIMLINE

MINI-FLOPPY DISK DRIVES

These half-height, 5¼" disk drives represent the state of the art in both technological design and mechanical construction. With the characteric high quality of manufacture expected in Japanese products, these drives feature ultra-modern electronics, servo-controlled direct drive motors and exceptional physical construction that provide extremely reliable, smooth and quiet operation in both single and double density. Two varieties will be available shortly: 40 track double sided and 80 track double sided.

Please WRITE for further information.

OTHER PRINTERS AVAILABLE:

EPSON RX-80 $995
Features: 100 cps, 6 character sizes, bit image and graphic modes.

ITOH PROWRITER 8510 $1150
Features: 120 cps, bit graphics and proportional printing.

EPSON FX-80 $1399
Features: 160 cps, 6 character sizes, proportional printing, bit graphics.

All prices include Sales Tax and are correct at time of publication but are subject to change without notice.
All equipment carries MICRO-80's Australia-wide 90 day warranty covering parts and labour.
Add $10 road freight anywhere in Australia.

```
540 S2=FIX(Y+(Y0-Q2)*8+.5):IFS2<
1 THEN S2=S2+8:Q2=Q2-1ELSE IFS2>
8THEN S2=S2-8:Q2=Q2+1
550 Q1= INT(Q1+.5):Q2= INT(Q2+.5
):IFQ1<1ORQ2<1ORQ1>8ORQ2>8THENQ1
=Q1-X1:Q2=Q2-X2:SD$=BL$+"EDGE OF
 GALAXY"+BL$:GOTO550
560 GOSUB860:GOTO190
570 IFK<1THEN SD$=BL$+"NO KLINGO
NS IN QUADRANT"+BL$:GOTO390ELSEQ
0$="":IFK>1THEN Q0$="S"
580 SD$=BL$+"PHASERS LOCKED ON K
LINGON"+Q0$+"  NO OF UNITS TO FI
RE?"+BL$:MX=1:PL=332:GOSUB990:X=
QQ:IFX<1THEN390ELSE IFE-X<1THEN5
80ELSE E=E-X
590 CLS:X=X/K:FORI=1TO3:IFK(I,3)
<1THEN620ELSE GOSUB950:PRINT
600 IFH<K(I,3)/10 THEN SX$=STR$(
K(I,1)):SY$=STR$(K(I,2)):SD$=BL$
+"SENSORS INDICATE NO DAMAGE ON
  KLINGON AT SECTOR "+SX$+":"+SY
$+BL$:GOTO620
610 K(I,3)=K(I,3)-H:HX$=STR$(H):
SX$=STR$(K(I,1)):SY$=STR$(K(I,2)
):SZ$=STR$(K(I,3)):SD$=BL$+HX$+"
 UNITS OF ENERGY HIT KLINGON AT
SECTOR "+SX$+":"+SY$+"  "+SZ$+"
LEFT"+BL$:IFK(I,3)<1 THEN SD$=SD
$+">>KLINGON DESTROYED<<"+BL$:GO
SUB930
620 NEXTI:GOSUB960:GOSUB860:GOTO
240
630 PC=0:IFP<1 THEN SD$=BL$+"BAD
 LUCK, YOUR OUT OF TORPEDOES"+BL
$:GOTO390ELSE IFP=1 THEN SD$=BL$
+"** LAST TORPEDO **"+BL$
640 MX=1:GOSUB420:SD$=BL$+"TORPE
DO COURSE (1-9)?"+BL$:PL=332:GOS
UB990:C=QQ:IFC<1THEN390ELSE IFC>
9THEN640
650 GOSUB760:X=S1:Y=S2:P=P-1
660 FORI=1TO300:NEXT:IFPC<>0 AND
 D(1)>0 THEN PRINT@PC-32,".";
670 X=X+X1:Y=Y+X2:IFX<.5 OR Y<.5
 OR X>=8.5 OR Y>=8.5 THEN SD$=BL
$+"BAD LUCK YOUR TORPEDO MISSED"
+BL$:GOTO740
680 X0=INT(X+.5):Y0=INT(Y+.5):PC
=(X0+1)*32+(Y0-1)+1:IFD(1)>0 THE
N PRINT@PC-32,CHR$(143);
690 IFQ(X0,Y0)=0THEN660ELSE IFD(
1)>0 THEN PRINT@PC-33,CHR$(134)"
 "CHR$(137);
700 X=X0:Y=Y0:IFQ(X,Y)=2 THEN FO
RI=1TO5:IFX=K(I,1) AND Y=K(I,2)
THEN K(I,3)=0:SD$=BL$+">>KLINGON
 DESTROYED<<"+BL$:GOSUB930:GOTO7
40ELSE NEXTI
710 IFQ(X,Y)=4 THEN SD$=BL$+"STA
R DESTROYED"+BL$:S=S-1:GOTO730
720 SD$=BL$+"STAR BASE DESTROYED
...          CONGRATULATIONS-TWI
T!!!"+BL$:B=0
730 Q(X,Y)=0:G(Q1,Q2)=K*100+B*10
+S
740 FORI=1TO500:NEXT:Q=0:GOSUB86
0:IFQ=1THEN250ELSE PRINT@114,C$"
 ";:PRINT@241,P" ";:PRINT@305,K9
;:IFD(1)>0 AND PC<>0 THEN PRINT@
PC-1,".";
750 IFD(2)< =0THEN390ELSE PRINT@
308,RIGHT$(" "+ STR$(G(Q1,Q2)),3
);:GOTO390
760 X2= COS((C-1)*.785398):X1= -
 SIN((C-1)*.785398):RETURN
770 MX=1:SD$=BL$+"ENERGY TO BE D
IVERTED TO SHIELDS?"+BL$:PL=332:
GOSUB990:D1=INT(QQ):IFE<=D1-D OR
 D1>500THEN390ELSE E=E-D1+D:D=D1
:PRINT@177,E"  ";:PRINT@241,D" "
;
780 PRINT@352,STRING$(21," ");:P
RINT@320,STRING$(21," "):GOTO390
790 CLS:PRINT@0,"DAMAGE REPORT":
PRINT@64,"DEVICE
 STATUS":PRINT@96,I$:FORI=0TO5:P
RINT@96+32*I,USING"%
    %      +##.#";D$(I),D(I):NEXT
:GOSUB970:GOTO240
800 FORI=1TO8:L$(I)="":NEXTI:CLS
:PRINT@0,USING"GALAXY MAP AT QUA
DRANT #:#";Q1,Q2:PRINT:GOSUB840:
FORI=1TO8:L$(I)=L$(I)+STR$(I)+"
":FORJ=1TO8:IFI=Q1 AND (J=Q2 OR
J=Q2+1) THEN L$(I)=L$(I)+CHR$(14
3)ELSE L$(I)=L$(I)+":"
810 LL=LEN(Z$(I,J)):IFLL<3THENZ$
(I,J)=STRING$(3-LL," ")+Z$(I,J)
820 L$(I)=L$(I)+" "+Z$(I,J)+" ":
NEXTJ:IFI=Q1 AND Q2=8 THENL$(I)=
L$(I)+CHR$(143)ELSE L$(I)=L$(I)+
":"
830 L$(I)=L$(I)+" "+STR$(I):NEXT
I:GOSUB850:GOSUB1390:GOTO240
840 L$(0)=" ":FORI=1TO8:L$(0)=L$
(0)+"    "+STR$(I):NEXTI:L$(0)=L
$(0)+" ":RETURN
850 L$(9)=" ":FORI=1TO8:L$(9)=L$
(9)+"    "+STR$(I):NEXTI:L$(9)=L
$(9)+" ":RETURN
860 IFK>0 THEN C$="RED" ELSEC$="
GREEN":IFE<350 THEN C$="YELLOW"
870 FORI=S1-1TOS1+1:FORJ=S2-1TOS
2+1:IFQ(I,J)<>3 THEN NEXTJ,I:GOT
O890
880 E=3510:P=10:D=510:FORI=0TO5:
D(I)=5:NEXT:C$="DOCKED":IFK>0 TH
EN CLS:Q=1:PRINT@448,"STARBASE S
HIELDS PROTECT          THE ENTERP
RISE";:FORI=1TO1000:NEXT:RETURN
890 IFK<1 THEN RETURN ELSEQ=1:CL
S:FORI=1TO3:X=K(I,3):IFX<=0THEN9
20ELSE GOSUB950:H=H+10:D=D-H:HX$
=STR$(H):SX$=STR$(K(I,1)):SY$=ST
R$(K(I,2)):SD$=BL$+HX$+" UNITS O
F ENERGY HIT ENTERPRISE FROM KLI
NGON AT SECTOR "+SX$+":"+SY$
900 K(I,3)=INT(X-X/4*RND(0)+.5):
IFD>0 THEN DX$=STR$(D):SD$=SD$+"
  SHIELDS REMAINING = "+DX$+BL$:
GOTO930
910 R=RND(6)-1:H1=INT((-D*(RND(0
)+.5)/50)*10+.5)/10:H=INT(-D-H1)
:D(R)=D(R)-H1:E=E-H:HX$=STR$(H):
EX$=STR$(E):HY$=STR$(H1)
920 SD$=" SHIELDS DOWN, YOU LOST
 "+HX$+"ENERGY POINTS."+EX$+" LE
FT & BEEN HIT ON YOUR "+D$(R)+"
FOR"+HY$+" POINTS.":NEXT:GOSUB96
0:IFE<1THEN1040ELSE RETURN
930 K=K-1:K9=K9-1:IFK9<1THEN1050
ELSEQ(K(I,1),K(I,2))=0:G(Q1,Q2)=
G(Q1,Q2)-100:RETURN
940 R1=RND(8):R2=RND(8):IFQ(R1,R
2)<>0THEN940ELSE RETURN
950 H= INT(X/(SQR((K(I,1)-S1)^2+
(K(I,2)-S2)^2))*(2+ RND(0))):RET
URN
960 RETURN
970 PRINT@480,"PRESS ANY KEY TO
CONTINUE";
980 IN$=INKEY$:IFIN$=""THEN980EL
SE CLS:RETURN
990 PRINT@PL,P$;:Q$="":QQ=0:Q1$=
INKEY$
1000 Q1$=INKEY$:GOSUB1450:IFQ1$=
""THEN1000ELSE Q=ASC(Q1$):IFQ=8T
HEN990ELSE IFQ<>13THEN IFQ<46 OR
 Q>57 OR Q=47THEN1000ELSEQ$=Q$+Q
1$:PRINT@PL,Q$;P$;:IF LEN(Q$)<6T
HEN1000
1010 IFQ$<>"" THENQQ=VAL(Q$)ELSE
QQ=0
1020 PRINT@PL-1,QQ"  ";:RETURN
1030 Q$= INKEY$:IFQ$< >"Y"ANDQ$<
 >"N"THEN1030ELSE RETURN
```

```
1040 FORI=1TO1000:NEXT:CLS:PRINT
@0," IT IS STARDATE"T" THE":PRIN
T"ENTERPRISE HAS BEEN DESTROYED
  THE FEDERATION WILL BE CONQUER
EDTHERE ARE STILL"K9"KLINGON":PR
INT"BATTLE CRUISERS LEFT ":PRINT
"...YOU ARE DEAD.":GOTO1100
1050 FORI=1TO1000:NEXT:CLS:PRINT
@0,"IT IS STARDATE"T:PRINT"THE L
AST KLINGON BATTLE CRUISER IN TH
E GALAXY HAS BEEN DESTROYEDTHE F
EDERATION HAS BEEN SAVED   YOU H
AVE BEEN PROMOTED TO       ADMIR
AL.  YOU DESTROYED"K0
1060 PRINT"KLINGONS IN "T-T0"STA
RDATES":F= INT(K0/(T-T0)*1000):0
NF/500+1GOTO1090,1080,1070,1070
1070 PRINT"YOUR RATING IS"F"WHIC
H IS       ABSOLUTELY FANTASTIC
I TAKE IT  YOU HAVE PLAYED BEFOR
E! THAT IS BETTER THAN ONE KLING
ON PER     STARDATE!":GOTO1100
1080 PRINT"NOT BAD, YOUR RATING
IS"F"IT    COULD BE BETTER":GOTO
1100
1090 PRINT"THIS IS A PRETTY LOUS
Y SCORE,   YOU CAN DO A LOT BETT
ER THAN    "F"!":GOTO1100
1100 PRINT@448,"DO YOU WISH TO A
TTEMPT IT AGAIN, (Y,N)?":GOSUB10
30:IFQ$="Y"THEN60ELSE PRINT"O.K.
 SEE YOU.":FORI=1TO1000:NEXT:CLS
:POKE359,126:END
1110 I$= CHR$(34):CLS:PRINT"THE
GALAXY IS DIVIDED INTO 64":PRINT
"QUADRANTS WITH THE FOLLOWING ":
PRINT"COORDINATES:-":PRINT"    1
 2 3 4 5 6 7 8":FORI=1TO8:PRINTI
;:FORJ=1TO8
1120 PRINT":";CHR$(45);:NEXT:PRI
NT":"I:NEXT:PRINT"    1 2 3 4 5
6 7 8":PRINT"EACH QUADRANT IS SI
MILARLY ":PRINT"DIVIDED INTO 64
SECTORS":GOSUB970
1130 PRINT TAB(5)"***DEVICES***"
:PRINT"COURSE (COMMAND 0):":PRIN
T"ANY REAL NUMBER BETWEEN 1 & 8.
9":PRINT"THE NUMBER INDICATING D
IRECTION STARTING AT THE RIGHT A
ND GOING COUNTER CLOCKWISE:"
1140 PRINT"        3":PRINT"
 4  :  2":PRINT"       * *":PRIN
T"    5---*---1":PRINT"       *
*":PRINT"     6  :  8":PRINT"
     7":GOSUB970
1150 PRINT"WARP ENGINES:":PRINT"
WARP FACTOR IS A REAL NUMBER ":P
RINT"BETWEEN 0 & 12 EACH WARP FA
CTOR":PRINT"WILL MOVE THE ENTERP
RISE ONE ":PRINT"QUADRANT. EACH
.125 WARP FACTOR":PRINT"WILL MOV
E THE ENTERPRISE ONE":PRINT"SECT
OR."
1160 PRINT:PRINT"EG: WARP .125 =
 1 SECTOR":PRINT"    WARP .5   =
 4 SECTORS":PRINT"    WARP  1
= 1 QUADRANT":PRINT"    WARP  5
  = 5 QUADRANTS":GOSUB970
1170 PRINT"FOR EXAMPLE, IF YOU T
RAVEL FROM:QUADRANT 1:1, SECTOR
1:1, IN    DIRECTION 1 AT WARP 2
 YOU WOULD STOP AT QUADRANT 1:2,
 SECTOR 1:1IN THE NEXT STARDATE.
"
1180 PRINT"":PRINTTAB(5)"** NOTE
 **":PRINT"EVERY USE OF THE WARP
 ENGINES   TAKES ONE STARDATE. I
F THE ":PRINT"ENTERPRISE IS BLOC
KED BY ":PRINT"SOMETHING DURING
INTRA-QUADRANT TRAVEL IT WILL ST
OP IN FRONT OF IT (AND WASTE A S
TARDATE)."
1190 GOSUB970:PRINT"SHORT RANGE
SENSORS:":PRINT"THE SHORT RANGE
SENSORS OF THE":PRINT"ENTERPRISE
 DISPLAY A DETAILED":PRINT"VIEW
OF THE QUADRANT IT IS IN"
1200 PRINT"THE ENTERPRISE LOOKS
LIKE 'E'":PRINT"ON THE SCREEN &
KLINGON BATTLE":PRINT"CRUISERS L
OOK LIKE 'A'":PRINT"STARBASES LO
OK LIKE '0' AND":PRINT"STARS LOO
K LIKE '*'":GOSUB970
1210 PRINT"LONG RANGE SENSORS:":
PRINT"THE LONG RANGE SENSORS OF
THE   ENTERPRISE DISPLAY INFORMA
TION  AS TO THE CONTENTS OF THE
NINE  CLOSEST QUADRANTS, THE CEN
TRE   BEING THE CURRENT QUADRANT
."
1220 PRINT"THE ONES DIGIT REPRES
ENT THE":PRINT"NUMBER OF STARS."
1230 PRINT" TENS            STARB
ASES        HUNDREDS        KLING
ONS."
1240 PRINT"FOR EXAMPLE:":PRINT"2
15 = 2 KLINGONS, 1 STARBASE &  5
 STARS. 108 = 1 KLINGON, 0     S
TARBASES AND 8 STARS. 2 = 0    K
LINGONS, 0 STARBASES & 2 STARS":
GOSUB970
1250 PRINT"GALAXY SCAN (COMMAND
5):":PRINT:PRINT"THIS COMMAND SH
OWS A CURRENT MAPOF THE KNOWN GA
LAXY. THIS IS    UPDATED BY THE
LONG RANGE       SCANNERS.":GOSU
B970
1260 PRINT"PHASERS (COMMAND 1):"
:PRINT:PRINT"ANY PORTION OF THE
ENERGY AVAIL-ABLE CAN BE FIRED,
THE ON BOARD BATTLE COMPUTER DIV
IDED THIS    AMOUNT AMONG THE KL
INGON        CRUISERS IN THE QUA
DRANT AND"
1270 PRINT"DETERMINES THE VARIOU
S          DIRECTIONS OF FIRE":P
RINT"THE EFFECTIVNESS OF A HIT I
S"
1280 PRINT"MAINLY DEPENDANT ON T
HE DISTANCETO THE CRUISER. EACH
CRUISER    STARTS WITH 200 UNITS
 OF ENERGY AND CAN FIRE AN AMOUN
T EQUAL TO HOWEVER MUCH IT HAS L
EFT.":GOSUB970
1290 PRINT"PHOTON TORPEDOES (COM
MAND 2):":PRINT
1300 PRINT"THE ENTERPRISE STARTS
 WITH 10   PHOTON TORPEDOES. ONE
 TORPEDO   WILL DESTROY WHATEVER
 IT HITS.  THE RANGE OF A PHOTON
 TORPEDO   (LIKE PHASERS) IS LIM
ITED TO"
1310 PRINT"THE CURRENT QUADRANT.
 THE COURSEOF A PHOTON TORPEDO I
S SET THE  SAME WAY AS THAT OF T
HE         ENTERPRISE":GOSUB970
1320 PRINT"DAMAGE CONTROL (COMMA
ND 4):":PRINT
1330 PRINT"THE DAMAGE CONTROL RE
PORT LISTS THE MAIN DEVICES AND
THEIR STATEOF REPAIR. A NEGATIVE
 STATE OF  REPAIR INDICATES A DI
SABLED     DEVICE. DEVICES CAN B
E DAMAGED"
1340 PRINT"BY A SPACE STORM OR K
LINGONS    AND ANY DAMAGED DEVIC
E IS       REPAIRED PARTIALLY EV
ERY        STARDATE.":GOSUB970
1350 PRINT"SHIELDS (COMMAND 3):"
:PRINT
1360 PRINT"SHIELDS WILL PROTECT
THE ENTERP-RISE FROM KLINGON FIR
E.   IF THESHIELDS DROP (TO 0)
THE KLINGONFIRE WILL DAMAGE & DI
SABLE SECT-IONS OF THE SHIP."
```

```
1370 PRINT"THE ENTERPRISE'S SHIE
LDS START  AT 500 UNITS (OF ENER
GY),  WHICHIS THE MAXIMUM.":GOSU
B970:CLS:GOTO90
1380 POKE359,57:SCREEN0,1:RETURN
1390 P=1
1400 PRINT@448,"PRESS ARROW KEYS
 TO MOVE THE MAPPRESS ENTER TO R
ETURN";:FORX=0TO9:PRINT@32*X+32,
MID$(L$(X),P,32);:NEXTX
1410 IN$=INKEY$:IFIN$=""THEN1410
 ELSE IN=ASC(IN$):IF IN=8THENP=P
-1:IFP=0THEN P=1
1420 IFIN=9THENP=P+1:IFP>24THENP
=P-1
1430 IFIN=13THENRETURN
1440 GOTO1400
1450 PRINT@352,MID$(SD$,MX,20);:
MX=MX+1:IFMX=LEN(SD$)-20THENMX=1
ELSEIFMX=255THENMX=1
1460 RETURN
```

```
      **** ARISTOCRAT ****

         COLOUR COMPUTER


10 REM

          POKER MACHINE

          ---------------


20 REM  (C) BOB DYBALL  16/02/81

          MODIFIED FOR THE COLOR
          COMPUTER  BY  MICRO-80

30 CLEAR1000:CLS:WAD=10:DIMR(3,3
0),RP(3):RP(1)=213:RP(2)=222:RP(
3)=230:CP$(1)="0":CP$(2)="0":CP$
(3)="@":CP$(4)="/":CP$(5)="-":CP
$(6)=" ":CP$(7)= CHR$(92):CP$(8)
="0":CP$(9)="0":CP$(10)="0"
40 :REM 'DRAW THE POKER MACHINE
THEN READ IN THE DATA FOR THE RE
ELS
50 CLS:GOSUB540:FORR=1TO3:FORP=1
TO30:READR(R,P):NEXTP:NEXTR
60 BL$=CHR$(128):BX$=STRING$(25,
BL$)+"press"+BL$+"the"+BL$+"spac
e"+BL$+"bar"+BL$+"to"+BL$+"inser
t"+BL$+"a"+BL$+"coin"+BL$+BL$+BL
$+"press"+BL$+"enter"+BL$+"for"+
BL$+"a"+BL$+"new"+BL$+"game"+STR
ING$(25,BL$)
70 PV=1
80 A$="":R=RND(30):S=RND(30):T=R
ND(30)
90 GOSUB690
100 PRINT@128,"";:PRINT USING"##
##";PAY;:IFWAD=0THEN670ELSE PRIN
T@32,"$";:PRINT USING"###.##";WA
D;:A$= INKEY$:IFA$=""THEN90ELSE
IFA$= CHR$(13)THEN CLS:GOTO30ELS
EWAD=WAD-.2:PRINT@32,"$";:PRINT
USING"###.##";WAD;
110 FORI=1TO14:ONI GOSUB160,160,
160,150,150,150,140,140,140,130,
140,150,160,170
120 NEXTI:GOTO190
130 FORD=1TO40:NEXTD:SOUND125,1:
RETURN
140 PRINT@156,"   ";:PRINT@188,"
  ";:PRINT@220,"   ";:PRINT@252,CH
R$(254)+CHR$(253);:PRINT@284,CHR
$(251)+CHR$(247);:PRINT@316,CHR$
(255)+"  ";:SOUND150,1:RETURN
150 PRINT@156,"   ";:PRINT@188,"
  ";:PRINT@220,CHR$(254)+CHR$(253
);:PRINT@252,CHR$(251)+CHR$(247)
;:PRINT@284,CHR$(255)+"  ";:SOUND
175,1:RETURN
160 PRINT@156,"   ";:PRINT@188,CH
R$(254)+CHR$(253);:PRINT@220,CHR
$(251)+CHR$(247);:PRINT@252,CHR$
(255)+"  ";:SOUND200,1:RETURN
170 PRINT@156,CHR$(254)+CHR$(253
);:PRINT@188,CHR$(251)+CHR$(247)
;:PRINT@220,CHR$(255)+"  ";:SOUND
225,1:RETURN
180 :REM '     DETERMINE APPROX.
HOW MANY TIMES EACH REEL TURNS O
VER         SOMETIMES THE REELS W
ILL TIP OVER JUST AFTER THEY STO
P
190 TK=RND(20):PAY=0:KK=RND(25)+
25:S1=(KK+RND(5)-3)*3/5:S2=(KK+R
ND(5)-3)*4/5
200 FOR RS=1TO KK:IF RS>S2 AND T
K=1 THEN240 ELSE IF RS>S2 THEN 2
70 ELSE IF RS>S1 THEN 240 ELSE R
=R-1:IFR<=1 THEN R=30
210 :REM ' IN 130-140,150-160,17
0-180 PRINT CHARACTERS ON THE RE
ELS
220 RA=R-1:RB=R-2:IFR=1THENRA=30
:RB=29
230 PRINT@172,CHR$(R(1,R));:PRIN
T@140,CHR$(R(1,RA));:PRINT@108,C
HR$(R(1,RB));:SOUND100,1
240 T=T-1:IFT<1THENT=30
250 TA=T-1:TB=T-2:IFT=1THENTA=30
:TB=29
260 PRINT@175,CHR$(R(2,T));:PRIN
T@143,CHR$(R(2,TA));:PRINT@111,C
HR$(R(2,TB));:SOUND110,1
270 S=S-1:IFS<1THENS=30
280 SA=S-1:SB=S-2:IFS=1THENSA=30
:SB=29
290 PRINT@178,CHR$(R(3,S));:PRIN
T@146,CHR$(R(3,SA));:PRINT@114,C
HR$(R(3,SB));:SOUND120,1
300 NEXT
310 :REM 'CHECK ON PAYOUT,IF LUC
KY, ADD TOGETHER THE 3 CHRS ON R
EELS
320 TR$=CHR$(R(1,RA))+CHR$(R(2,T
A))+CHR$(R(3,SA))
330 :REM 'CHECK IF JACKPOT
340 IFTR$=CHR$(128)+CHR$(128)+CH
R$(128)THEN PRINT@453,"JACKPOT -
 $50 -       ";:FORD=1TO100:NEXT:
PAY=50:GOTO480
350 IF LEFT$(TR$,2)=CHR$(128)+CH
R$(128)THEN PRINT@453,"JACKPOT -
 $20 -       ";:FORD=1TO100:NEXT:
PAY=20:GOTO480
360 :REM 'CHECK IF RUN
370 IFTR$=CHR$(159)+CHR$(207)+CH
R$(175)ORTR$=CHR$(175)+CHR$(191)
+CHR$(128)ORTR$=CHR$(207)+CHR$(1
75)+CHR$(191)THENPAY=4:GOTO470
380 X$=CHR$(R(1,RA)):Y$=CHR$(R(2
,TA)):Z$=CHR$(R(3,SA))
390 :REM '   CHECK IF A SMALLER
PAYOUT
400 IFX$=Y$ANDY$=Z$ANDZ$=CHR$(15
9)THENPAY=2:GOTO470
410 IFX$=Y$ANDX$=CHR$(159)THENPA
Y=1:GOTO470
420 IFX$=Y$ANDY$=Z$ANDZ$=CHR$(20
7)THENPAY=3:GOTO470
430 IFX$=Y$ANDY$=CHR$(207)THENPA
Y=2:GOTO470
440 IFX$=CHR$(159)THENPAY=.2:GOT
O470
450 IFX$=Y$ANDY$=Z$THENPAY=6:GOT
O470ELSE100::REM 'GOTO 40 IF NO
PAYOUT
460 :REM ADJUST WAD
```

```
470 WAD=WAD+PAY:FORI=1TO PAY:A=R
ND(60)+60:NEXT:PAY=PAY*5:GOSUB68
0:GOTO100
480 WAD=WAD+PAY:PAY=PAY*5:GOSUB6
80:GOTO100
490 :REM 'DATA FOR THE THREE REE
LS
500 DATA159,175,191,128,207,175,
159,191,159,191,175,191,207,159,
207,159,175,191,191,207,159,159,
175,191,175,191,207,191,175,191
510 DATA191,175,207,159,128,159,
207,128,175,191,175,159,175,191,
175,207,159,159,207,159,191,159,
207,175,175,159,191,159,159,191
520 DATA159,207,175,159,128,191,
175,159,207,159,175,191,128,159,
159,175,191,191,128,175,159,191,
207,175,159,207,175,159,207,175
530 :REM 'DRAW  THE POKER MACHIN
E ,PUT IN THE PAYOUT LISTING.
540 'DRAW MACHINE
550 PRINT@347,CHR$(255)+CHR$(255
);:PRINT@316,CHR$(255);:PRINT@28
4,CHR$(255);:PRINT@252,CHR$(255)
;:GOSUB170
560 FORX=2OTO41:SET(X,0,8):SET(X
,1,8):SET(X,4,8):SET(X,5,8):SET(
X,12,8):SET(X,13,8):NEXT:FORY=OT
O13:SET(18,Y,8):SET(19,Y,8):SET(
42,Y,8):SET(43,Y,8):NEXT
570 FORY=6TO11:SET(20,Y,8):SET(2
1,Y,8):SET(22,Y,8):SET(23,Y,8):S
ET(26,Y,8):SET(27,Y,8):SET(28,Y,
8):SET(29,Y,8):SET(32,Y,8):SET(3
3,Y,8):SET(34,Y,8):SET(35,Y,8):S
ET(38,Y,8):SET(39,Y,8):SET(40,Y,
8):SET(41,Y,8):NEXT
580 SET(22,8,6):SET(23,8,6):SET(
26,8,6):SET(29,8,6):SET(32,8,6):
SET(33,8,6):SET(34,8,6):SET(35,8
,6):SET(38,8,6)
590 FORX=8TO20:SET(X,12,8):SET(X
+33,12,8):SET(X,13,8):SET(X+33,1
3,8):NEXT:FORY=14TO31:SET(8,Y,8)
:SET(9,Y,8):SET(52,Y,8):SET(53,Y
,8):NEXT:FORX=8TO53:SET(X,24,8):
SET(X,25,8):SET(X,26,8):SET(X,27
,8):SET(X,30,8):SET(X,31,8):NEXT
600 PRINT@42,"aristocrat";:PRINT
CHR$(128);:PRINT@108,CHR$(207);:
PRINT@111,CHR$(175);:PRINT@114,C
HR$(191);:PRINT@140,CHR$(159);:P
RINT@143,CHR$(128);:PRINT@146,CH
R$(175);:PRINT@172,CHR$(128);:PR
INT@175,CHR$(159);
610 PRINT@178,CHR$(175);:PRINT@0
,"credit";:PRINT@32,"$";:PRINT U
SING"###.##";WAD;:PRINT@229,CHR$
(159);" x x :01    ";CHR$(175);"
";CHR$(175);" ";CHR$(175);" :30"
;:PRINT@261,CHR$(159);" ";CHR$(1
59);" x :05    ";
620 PRINTCHR$(191);" ";CHR$(191)
;" ";CHR$(191);" :30";
630 PRINT@293,CHR$(159);" ";CHR$
(159);" ";CHR$(159);" :10   r u
n :20";:PRINT@325,CHR$(207);" ";
CHR$(207);" x :10    ";CHR$(128);
" ";CHR$(128);" x $20";:PRINT@35
7,CHR$(207);" ";CHR$(207);" ";CH
R$(207);" :30    ";CHR$(128);" ";
CHR$(128);" ";
640 PRINTCHR$(128);
650 PRINT" $50";:PRINT@96,"paid"
;:RETURN
660 :REM 'IF YOU RUN OUT OF MONE
Y
670 CLS:PRINT@256,"YOU CANT PLAY
 WITHOUT MONEY!":PRINT@288,"YOU
HAVE JUST BEEN BOOTED OUT":PRINT
@324," OF THE CLUB!":END
680 FOR NT=1 TO PAY:SOUND230,1:F
OR NI=1TO25:NEXT NI:NEXT NT:RETU
RN
690 PRINT@453,MID$(BX$,PV,21);:P
RINT@128,"";:PV=PV+1:IFPV=90THEN
PV=1
700 RETURN
```

```
**** STAR TREK ****

HITACHI PEACH


10 ' **     STAR TREK     **
20 ' ** R.O. EDWARDS      **
30 ' ** 31 REDESDALE RD **
40 ' ** IVANHOE     3079 **
50 ' ** MODIFIED FOR THE HITACHI PEACH B
Y MICRO-80 **
60 RANDOMIZE:DEFINTI,J,G,K,Q:CLEAR800:RE
STORE:CLS:LOCATE25,9:PRINT"STAR TREK":FO
RI=1TO1000:NEXT:DEF FNR(RA)=INT(RND(1)*M
AX)+1
70 CLS:SCREEN0,,1:LOCATE19,9:PRINT"DO YO
U REQUIRE INSTRUCTIONS, (Y,N)?":GOSUB102
0:IFQ$="Y"THEN1100
80 LOCATE19,10:PRINT"NO OF KLINGONS, ( 1
(LOW)-5(HIGH) )";:GOSUB1320:IFN<1ORN>5TH
EN80ELSECLS:LOCATE18,10:PRINT"CREATING G
ALAXY":Q=1.8^N
90 K9=0:B1=0:K1=0:MAX=20:T0=FNR(RA):T0=T
0+20:T0=T0*100:T=T0:FORI=1TO8:FORJ=1TO8:
MAX=500:R=FNR(RA):R=R/Q
100 K=0:IFR<20THENK=1:K1=K1+1:IFR<10THEN
K=2:IFR<5THENK=3
110 K9=K9+K:B=0:IFRND(1)>1-N/50THENB=10:
B1=1
120 MAX=9:R9=FNR(RA):G(I,J)=K*100+B+R9:Z
$(I,J)="   ":NEXTJ,I:K0=K9:IFK1>60ORK1<3
THEN90
130 MAX=8:Q1=FNR(RA):Q2=FNR(RA):IFINT(G(
Q1,Q2)/100)>0THEN130ELSEIFB1<>1THEN MAX=
8:I=FNR(RA):J=FNR(RA):G(I,J)=G(I,J)+10
140 CLS:T9=K9*3:MAX=8:S1=FNR(RA):S2=FNR(
RA):LOCATE7,6:PRINT"PRESENT STARDATE"T"A
S COMMANDER OF THE U.S.S. ENTERPRISE YOU
R MISSION IS TO RID THE GALAXY OF THE  D
EADLY KLINGON MENACE, TO DO THIS, YOU MU
ST DESTROY THE KLINGON INVASION FORCE"
150 PRINT"OF ";K9;" BATTLE CRUISERS":PRI
NT:PRINT"YOU HAVE"T9"SOLAR YEARS TO COMP
LETE YOUR MISSION(I.E. UNTIL STARDATE"T0
+T9")":GOSUB960
160 FORI=0TO5:READD$(I):D(I)=5:NEXT:E=35
00:D=500:P=10:P$=CHR$(95)+"      ":I$=STR
ING$(21,45):DATAWARP ENGINES,SECTOR SCAN
NER,DEEP SPACE SCANNER,PHASER CONTROL,PH
OTON TUBES,SHIELD CONTROL
170 X=G(Q1,Q2)/100:K=INT(X):B=INT((X-K)*
10):S=G(Q1,Q2)-INT(G(Q1,Q2)/10)*10
180 FORI=1TO8:FORJ=1TO8:Q(I,J)=0:NEXTJ,I
:FORI=1TO3:K(I,3)=0:NEXT
190 Q(S1,S2)=1:IFK>0THENFORI=1TOK:GOSUB9
40:Q(R1,R2)=2:K(I,1)=R1:K(I,2)=R2:K(I,3)
=200:NEXT
200 IFB>0THENGOSUB940:Q(R1,R2)=3
210 GOSUB860:IFS>0THENFORI=1TOS:GOSUB940
:Q(R1,R2)=4:NEXT
220 IFD(2)>0THENFORI=Q1-1TOQ1+1:FORJ=Q2-
1TOQ2+1:Z$(I,J)=RIGHT$(" "+STR$(G(I,J)),
3):NEXTJ,I
230 CLS:PRINT"       SHORT RANGE SCAN"TA
B(54)"LONG RANGE SCAN":LOCATE7,4:PRINTI$
;I$" "I$:FORI=1TO8:IFD(1)>0THENFORJ=1TO8
:PRINT" "MID$(".EAO*",Q(I,J)+1,1);:NEXTJ
240 PRINTTAB(27);
250 ON I GOSUB300,310,320,330,340,350,36
0,370
260 NEXTI
270 PRINTI$;I$
280 IFD(1)<=0THENLOCATE8,8:PRINT"** INOP
ERABLE **";
```

```
290 GOTO400
300 PRINT"STARDATE          "T;:PRINT TAB(
51):Q=Q1-1:GOSUB380:RETURN
310 PRINT"CONDITION         "C$;:PRINTTAB
(51)I$:RETURN
320 PRINTUSING"QUADRANT           #:#";Q1,
Q2;:PRINTTAB(51):Q=Q1:GOSUB380:RETURN
330 PRINTUSING"SECTOR             #:#";S1,
S2;:PRINTTAB(51)I$:RETURN
340 PRINT"ENERGY            "E;:PRINT TAB(
51):Q=Q1+1:GOSUB380:RETURN
350 PRINT"PHOTON TORPEDOES"P;:PRINT TAB(
51)I$:RETURN
360 PRINT"SHIELDS           "D:RETURN
370 PRINT"KLINGONS LEFT     "K9:RETURN
380 IFD(2)>0 THEN PRINTUSING" * ### * ##
# * ### *";G(Q,Q2-1),G(Q,Q2),G(Q,Q2+1) E
LSE IF Q=Q1 THEN PRINT"   ** INOPERABLE *
*" ELSE PRINT
390 RETURN
400 LOCATE7,14:PRINT"
            ":LOCATE7,15:PRINT"
                      ":LOCATE7,16:PRINT
"                           ":LOCATE
7,14:PRINT"COMMAND?";:GOSUB420:Z$(Q1,Q2)
=RIGHT$(" "+STR$(G(Q1,Q2)),3)
410 Q$=INKEY$:IFQ$=""THEN410ELSEA=ASC(Q$
)-48:IFA<0ORA>5THEN410ELSEIFA<4ANDA>0AND
D(A+2)<=0THENLOCATE7,14:PRINT"** "D$(A+2
)" INOPERABLE **";:FORI=1TO1000:NEXT:GOT
0400ELSE ON A+1 GOTO460,600,660,800,810,
820
420 LOCATE55,12:PRINT"COMMANDS:";:LOCATE
52,13:PRINT"0 SET COURSE";:LOCATE52,14:P
RINT"1 PHASERS ";:LOCATE52,15:PRINT"2 PH
OTON TORPEDOES";:LOCATE52,16:PRINT"3 SHI
ELDS ";:LOCATE52,17:PRINT"4 DAMAGE REPOR
T";
430 LOCATE52,18:PRINT"5 GALAXY MAP";RETU
RN
440 LOCATE55,12:PRINT"      3    ";:LOCATE
52,13:PRINT"      4   :   2";:LOCATE52,14:P
RINT"        *:*";:LOCATE52,15:PRINT"
5---*---1       ";:LOCATE52,16:PRINT"
  *:*";:LOCATE52,17:PRINT"      6   :   8
 ";:LOCATE52,18:PRINT"         7    ";
450 RETURN
460 GOSUB440:LOCATE7,14:PRINT"COURSE (1-
9)?";:PL=718:GOSUB980:C=QQ:IFC<1THEN400E
LSEIFC>9THEN460
470 LOCATE7,15:PRINT"WARP FACTOR (.1-12)
?";:PL=789:GOSUB980:W=QQ:IFW<=0THEN400EL
SEIFW>12THEN470
480 IFW>1ANDD(0)<=0THENLOCATE7,16:PRINT"
WARP ENGINES DAMAGED, MAX SPEED WARP 1";
:FORI=1TO1000:NEXT:GOTO400
490 CLS:T=T+1:IFT>T0+T9THEN1000ELSEFORI=
0TO5:D(I)=D(I)+.5:IFD(I)>5THEND(I)=5ELSE
IFD(I)<-5THEND(I)=D(I)+.5
500 NEXT:MAX=50:RX=FNR(RA):MAX=1100:RY=F
NR(RA):IFRX<W THEN D=D-RY-200:LOCATE14,1
0:PRINT"*** SPACE STORM *** ";:IFD>0 THE
N PRINT"SHIELDS HELD ***"ELSE MAX=6:R=FN
R(RA):R=R-1:D(R)=D(R)+D/100:PRINTD$(R);"
 DAMAGED ***":D=0
510 N=INT(W*8):E=INT(E-N*2):IFE<1THEN103
0ELSEQ(S1,S2)=0:X=S1:Y=S2:GOSUB790:FORI=
1TON:S1=S1+X1:S2=S2+X2
520 IFS1<.5ORS2<.5ORS1>=8.5ORS2>=8.5THEN
560
530 IFQ(INT(S1+.5),INT(S2+.5))>1THEN LOC
ATE19,10:PRINTUSING"ENTERPRISE IS BLOCKE
D BY OBJECT AT SECTOR #:#";S1,S2:S1=INT(
S1-X1+.5):S2=INT(S2-X2+.5):FORI=1TO1000:
NEXT:GOTO550
540 NEXTI:S1=INT(S1+.5):S2=INT(S2+.5)
550 Q(S1,S2)=1:GOSUB860:GOTO220
560 X0=Q1+W*X1:Q1=FIX(X0):Y0=Q2+W*X2:Q2=
FIX(Y0):S1=FIX(X+(X0-Q1)*8+.5):IFS1<1THE
N S1=S1+8:Q1=Q1-1ELSEIFS1>8THEN S1=S1-8:
Q1=Q1+1
570 S2=FIX(Y+(Y0-Q2)*8+.5):IFS2<1THEN S2
=S2+8:Q2=Q2-1ELSEIFS2>8THEN S2=S2-8:Q2=Q
2+1
580 Q1=INT(Q1+.5):Q2=INT(Q2+.5):IFQ1<1 O
R Q2<1 OR Q1>8 OR Q2>8THENQ1=Q1-X1:Q2=Q2
-X2:LOCATE30,11:PRINT"EDGE OF GALAXY":GO
TO580
590 GOSUB860:GOTO170
600 IFK<1THENLOCATE7,14:PRINT"NO KLINGON
S IN QUADRANT!!";:FORI=1TO1000:NEXT:GOTO
400ELSEQ0$="":IFK>1THENQ0$="S"
610 LOCATE7,14:PRINT"PHASERS LOCKED ON K
LINGON"Q0$","";:LOCATE7,15:PRINT"NO OF UN
ITS TO FIRE?";:PL=789:GOSUB980:X=QQ:IFX<
1THEN400ELSEIFE-X<1THEN610ELSEE=E-X
620 CLS:X=X/K:FORI=1TO3:IFK(I,3)<1THEN65
0ELSEGOSUB950:PRINT
630 IFH<K(I,3)/10THEN PRINTUSING"
SENSORS INDICATE NO DAMAGE ON KLINGON AT
 SECTOR #:#";K(I,1),K(I,2):GOTO650
640 K(I,3)=K(I,3)-H:PRINTUSING"#,### UNI
TS OF ENERGY HIT KLINGON AT SECTOR #:#.
(##,### LEFT)";H,K(I,1),K(I,2),K(I,3):IF
K(I,3)<1THENPRINTTAB(23)">>KLINGON DESTR
OYED<<":GOSUB930
650 NEXTI:GOSUB960:GOSUB860:GOTO220
660 PC=0:IFP<1THEN LOCATE7,14:PRINT"BAD
LUCK, YOUR OUT OF TORPEDOES";:FORI=1TO10
00:NEXT:GOTO400ELSEIFP=1THEN LOCATE7,15:
PRINT"** LAST TORPEDO **";
670 GOSUB440:LOCATE7,14:PRINT"TORPEDO CO
URSE (1-9)?";:PL=726:GOSUB980:C=QQ:IFC<1
THEN400ELSEIFC>9THEN670
680 GOSUB790:X=S1:Y=S2:P=P-1
690 FORI=1TO300:NEXT:IFPC<>0 AND D(1)>0T
HEN POKE1344+PC-80,46
700 X=X+X1:Y=Y+X2:IFX<.5ORY<.5ORX>=8.5OR
Y>=8.5THEN LOCATE7,16:PRINT"BAD LUCK YOU
R TORPEDO MISSED";:GOTO770
710 X0=INT(X+.5):Y0=INT(Y+.5):PC=(X0+1)*
80+(Y0-1)*2+1:IFD(1)>0THEN POKE1344+PC-8
0,140
720 IFQ(X0,Y0)=0THEN690ELSEIFD(1)>0THENP
OKE1344+PC-1,140
730 X=X0:Y=Y0:IFQ(X,Y)=2THENFORI=1TO5:IF
X=K(I,1)ANDY=K(I,2)THENK(I,3)=0:LOCATE7,
16:PRINT">>KLINGON DESTROYED<<";:GOSUB93
0:GOTO770ELSENEXTI
740 IFQ(X,Y)=4THENLOCATE7,16:PRINT"STAR
DESTROYED";:S=S-1:GOTO760
750 LOCATE7,16:PRINT"STAR BASE DESTROYED
...CONGRATULATIONS-TWIT!!!";:B=0
760 Q(X,Y)=0:G(Q1,Q2)=K*100+B*10+S
770 FORI=1TO500:NEXT:Q=0:GOSUB860:IFQ=1T
HEN230ELSELOCATE43,6:PRINTC$" ";:LOCATE4
2,10:PRINTP" ";:LOCATE42,12:PRINTK9;:IFD
(1)>0ANDPC<>0THENPOKE1344+PC-80,46
780 IFD(2)<=0THEN400ELSELOCATE59,7:PRINT
RIGHT$(" "+STR$(G(Q1,Q2)),3);:GOTO400
790 X2=COS((C-1)*.785398):X1=-SIN((C-1)*
.785398):RETURN
800 LOCATE7,14:PRINT"ENERGY TO BE DIVERT
ED":LOCATE7,15:PRINT"TO SHIELDS?";:PL=73
8:GOSUB980:D1=INT(QQ):IF E<=D1-D OR D1>5
00 THEN400ELSE E=E-D1+D:D=D1:LOCATE41,9:
PRINT"  "E" ";:LOCATE43,11:PRINTD" ":GOT
0400
810 CLS:LOCATE31,4:PRINT"DAMAGE REPORT":
LOCATE23,6:PRINT"DEVICE
STATUS":LOCATE23,7:PRINTI$"--------":FOR
I=0TO5:LOCATE23,I+8 :PRINTUSING"&
          &      +##.#";D$(I),D(I):NEXT:GO
SUB960:GOTO220
820 CLS:LOCATE27,3:PRINTUSING"GALAXY MAP
 AT QUADRANT #:#";Q1,Q2:PRINT:GOSUB850:F
ORI=1TO8:PRINTTAB(13)I" ";:FORJ=1TO8:IF
I=Q1 AND (J=Q2 OR J=Q2+1)THENPRINTCHR$(1
40);ELSEPRINT":";
830 PRINT" "Z$(I,J)" ";:NEXTJ:IF I=Q1 AN
D Q2=8THENPRINTCHR$(140);ELSEPRINT":";
840 PRINT" "I:NEXTI:GOSUB850:GOSUB960:GO
TO220
850 PRINTTAB(16);:FORI=1TO8:PRINT"   "I;
:NEXTI:PRINT:RETURN
860 IFK>0THENC$="RED"ELSEC$="GREEN":IFE<
350THENC$="YELLOW"
```

```
870 FORI=S1-1TOS1+1:FORJ=S2-1TOS2+1:IFQ(
I,J)<>3THENNEXTJ,I:GOTO890
880 E=3500:P=10:D=500:FORI=0TO5:D(I)=5:N
EXT:C$="DOCKED":IFK>0THENCLS:Q=1:LOCATE1
9,10:PRINT"STARBASE SHIELDS PROTECT THE
ENTERPRISE";:FORI=1TO1000:NEXT:RETURN
890 IFK<1THENRETURNELSEQ=1:CLS:FORI=1TO3
:X=K(I,3):IFX<=0THEN920ELSEGOSUB950:H=H+
10:D=D-H:PRINT:PRINTUSING"#,### UNITS OF
 ENERGY HIT ENTERPRISE FROM KLINGON AT S
ECTOR #:#";H,K(I,1),K(I,2)
900 K(I,3)=INT(X-X/4*RND(1)+.5):IF D>0 T
HEN PRINTTAB(23);:PRINTUSING"SHIELDS REM
AINING=###";D:GOTO920
910 MAX=6:R=FNR(RA):R=R-1:H1=INT((-D*(RN
D(1)+.5)/50)*10+.5)/10:H=INT(-D-H1):D(R)
=D(R)-H1:E=E-H:PRINTUSING"SHIELDS ARE DO
WN, YOU HAVE LOST ### ENERGY POINTS (#,#
## LEFT)";H,E:PRINT"AND BEEN HIT ON YOUR
 "D$(R)" FOR"H1"POINTS.":D=0
920 NEXT:GOSUB960:IFE<1THEN1030ELSERETUR
N
930 K=K-1:K9=K9-1:IFK9<1THEN1040ELSEQ(K(
I,1),K(I,2))=0:G(Q1,Q2)=G(Q1,Q2)-100:RET
URN
940 MAX=8:R1=FNR(RA):R2=FNR(RA):IFQ(R1,R
2)<>0THEN940ELSERETURN
950 H=INT(X/(SQR((K(I,1)-S1)^2+(K(I,2)-S
2)^2))*(2+RND(0))):RETURN
960 LOCATE7,18:PRINT"PRESS ANY KEY TO CO
NTINUE";:Q$=INKEY$
970 IFINKEY$=""THEN970ELSECLS:RETURN
980 LOCATE35,14:Q$="":QQ=0:Q1$=INKEY$
990 Q1$=INKEY$:IFQ1$=""THEN990ELSEQ=ASC(
Q1$):IFQ=8THEN980ELSEIFQ<>13THENIFQ<46OR
Q>57ORQ=47THEN990ELSEQ$=Q$+Q1$:LOCATE35,
14:PRINTQ$;P$;:IFLEN(Q$)<6THEN990
1000 IFQ$<>""THENQQ=VAL(Q$)ELSEQQ=0
1010 LOCATE35,14:PRINT"      ";:RETURN
1020 Q$=INKEY$:IFQ$<>"Y"ANDQ$<>"N"THEN10
20ELSERETURN
1030 FORI=1TO1000:NEXT:CLS:LOCATE7,8:PRI
NT" IT IS STARDATE"T" THE ENTERPRISE HAS
 BEEN DESTROYED THE FEDERATION WILL BE C
ONQUERED THERE ARE STILL"K9"KLINGON BATT
LE CRUISERS LEFT     ...YOU ARE DEAD.":GO
TO1090
1040 FORI=1TO1000:NEXT:CLS:LOCATE7,6:PRI
NT"         IT IS STARDATE"T"THE LAST KLI
NGON BATTLE CRUISER IN THE GALAXY HAS BE
EN DESTROYEDTHE FEDERATION HAS BEEN SAVE
DYOU HAVE BEEN PROMOTED TO ADMIRAL.YOU D
ESTROYED"K0"KLINGONS IN"T-T0"STARDATES"
1050 F=INT(K0/(T-T0)*1000):ONF/500+1GOTO
1080,1070,1060,1060
1060 PRINT"YOUR RATING IS"F"WHICH IS ABS
OLUTELY FANTASTICI TAKE IT YOU HAVE PLAY
ED BEFORE!THAT IS BETTER THAN ONE KLINGO
N PER STARDATE!":GOTO1090
1070 PRINT"NOT BAD, YOUR RATING IS"F"IT
COULD BE BETTER":GOTO1090
1080 PRINT"THIS IS A PRETTY LOUSY SCORE,
 YOU CAN DO AHELL OF A LOT BETTER THAN"F
"!":GOTO1090
1090 LOCATE7,16:PRINT"DO YOU WISH TO ATT
EMPT IT AGAIN, (Y,N)?":GOSUB1020:IFQ$="Y
"THEN60ELSEPRINT"O.K. SEE YOU.":FORI=1TO
1000:NEXT:CLS:END
1100 I$=CHR$(34):CLS:PRINT"THE GALAXY IS
 DIVIDED INTO 64 QUADRANTS WITH THE FOLL
OWING COORDINATES:":PRINT"     1   2   3
   4   5   6   7   8":FORI=1TO8:PRINTI;:
FORJ=1TO8
1110 PRINT":"STRING$(3,95);:NEXT:PRINT":
"I:NEXT:PRINT"     1   2   3   4   5   6
   7   8":PRINT"EACH QUADRANT IS SIMILAR
LY DIVIDED INTO 64 SECTORS":GOSUB960
1120 PRINTTAB(33)"***DEVICES***":PRINT"C
OURSE (COMMAND 0):":PRINT"ANY REAL NUMBE
R BETWEEN 1 AND 8.9, THE NUMBER INDICATI
NG DIRECTION STARTING AT  THE RIGHT AND
GOING COUNTER CLOCKWISE:"
1130 PRINT"         3    ":PRINT"     4
:  2 ":PRINT"    5---*---1":PRINT"     6
  :  8 ":PRINT"         7":GOSUB960
1140 PRINT"WARP ENGINES:":PRINT:PRINT"WA
RP FACTOR IS A REAL NUMBER BETWEEN 0 AND
 12  EACH WARP FACTOR WILL MOVE THE   EN
TERPRISE ONE QUADRANT. EACH .125 WARP FA
CTORS WILL MOVE THE ENTERPRISE ONE    SE
CTOR."
1150 PRINT"EG: WARP .125 = 1 SECTOR":PRI
NT"    WARP .5   = 4 SECTORS (HALF A QUA
DRANT":PRINT"    WARP  1   = 1 QUADRANT"
:PRINT"    WARP  5   = 5 QUADRANTS":GOSU
B960
1160 PRINT:PRINT"FOR EXAMPLE, IF YOU TRA
VEL FROM:":PRINT"QUADRANT 1:1, SECTOR 1:
1, IN DIRECTION 1 AT WARP 2 YOU WOULD ST
OP AT":PRINT"QUADRANT 1:3, SECTOR 1:1, I
N THE NEXT STARDATE."
1170 PRINT""TAB(23)"** NOTE **":PRINT"EV
ERY USE OF THE WARP ENGINES TAKES ONE ST
ARDATE.  IF THE ENTERPRISE IS BLOCKED BY
 SOMETHING DURING INTRA-QUADRANT TRAVEL
IT WILL STOP IN FRONT OF IT          (A
ND WASTE A STARDATE).":GOSUB960
1180 PRINT"SHORT RANGE SENSORS:":PRINT"T
HE SHORT RANGE SENSORS OF THE ENTERPRISE
 DISPLAYS A DETAILED VIEW OF THE      Q
UADRANT IT IS CURRENTLY IN."
1190 PRINT"THE ENTERPRISE LOOKS LIKE "I$
"E"I$"ON THE SCREEN AND KLINGON BATTLE C
RUISERS LOOK LIKE "I$"A"I$",STARBASES LO
OK LIKE "I$"O"I$"AND STARS LOOK LIKE "I$
"*"I$:GOSUB960
1200 PRINT"LONG RANGE SENSORS:":PRINT:PR
INT"THE LONG RANGE SENSORS OF THE ENTERP
RISE DISPLAYS INFORMATION AS TO THE CONT
ENTSOF THE NINE CLOSEST QUADRANTS, THE C
ENTRE BEING THE CURRENT QUADRANT."
1210 PRINT"THE ONES DIGIT REPRESENTS THE
 NUMBER OF STARS.":PRINT"TENS
                    STARBASES.":PRINT"H
UNDREDS                          KLING
ONS."
1220 PRINT"FOR EXAMPLE:":PRINT"215 MEANS
 2 KLINGONS, 1 STARBASE,  AND 5 STARS.":
PRINT"108 MEANS 1 KLINGON,  0 STARBASES,
 AND 8 STARS.":PRINT"2 MEANS 0 KLINGONS,
 0 STARBASES, AND 2 STARS.":GOSUB960
1230 PRINT"GALAXY SCAN (COMMAND 5):":PRI
NT:PRINT"THIS COMMAND SHOWS A CURRENT MA
P OF THE KNOWN GALAXY.      THIS IS UPDAT
ED BY THELONG RANGE SCANNERS.":GOSUB960
1240 PRINT"PHASERS (COMMAND 1):":PRINT:P
RINT"ANY PORTION OF THE ENERGY AVAILABLE
 CAN BE FIRED,   THE ON-BOARD BATTLE COM
PUTERDIVIDES THIS AMOUNT AMONG THE KLING
ON CRUISERS IN THE QUADRANT AND DETERMIN
ES   THE VARIOUS DIRECTIONS OF FIRE."
1250 PRINT"THE EFFECTIVENESS OF A HIT IS
 MAINLY DEPENDANT ON THE DISTANCE TO THE
 CRUISER.  EACH CRUISER STARTS WITH 200
UNITS OF ENERGY AND CAN FIRE AN AMOUNT E
QUAL TO    HOWEVER MUCH IT HAS LEFT.":GO
SUB960
1260 PRINT"PHOTON TORPEDOES (COMMAND 2):
":PRINT:PRINT"THE ENTERPRISE STARTS WITH
 10 PHOTON TORPEDOES, ONE TORPEDO DESTRO
YS WHATEVER ITHITS. THE RANGEOF A PHOTON
 TORPEDO (LIKE PHASERS) IS LIMITED TO"
1270 PRINT"THE CURRENT QUADRANT. THE COU
RSE OF A PHOTON TORPEDO IS SET THE SAME
WAY AS THATOF THE ENTERPRISE.":GOSUB960
1280 PRINT"DAMAGE CONTROL REPORT (COMMAN
D 4):":PRINT:PRINT"THE DAMAGE CONTROL RE
PORT LISTS THE MAIN DEVICES AND THEIR ST
ATE OF REPAIR.      A NEGATIVE STATE OF R
EPAIR INDICATES A DISABLED DEVICE. DEVIC
ES CAN BE DAMAGED"
1290 PRINT"BY A SPACE STORM OR KLINGONS,
 AND ANY DAMAGED DEVICE IS REPAIRED PART
IALLY EVERY STARDATE.":GOSUB960
```

```
1300 PRINT"SHIELDS (COMMAND 3):":PRINT:P
RINT"SHIELDS WILL PROTECT THE ENTERPRISE
 FROM KLINGON FIRE. IF SHIELDS DROP (TO
0)THEKLINGON FIRE WILL DAMAGE AND DISABL
E SECTIONS OF THE SHIP."
1310 PRINT"THE ENTERPRISE'S SHIELDS STAR
T AT 500 UNITS (OF ENERGY), WHICH IS THE
 MAXIMUM.":GOSUB960:CLS:GOTO80
1320 N$=INKEY$:IFN$=""THEN1320ELSEN=VAL(
N$):RETURN
```

****** ARISTOCRAT ******

HITACHI PEACH

```
10 ' (C) R. DYBALL   : MODIFIED FOR THE
HITACHI PEACH BY MICRO-80
20 RANDOMIZE:CLEAR1000:DEFINTA-O,Q-V,X-Z
:DEF FNRN(RA)=INT(RND(1)*MAX)+1
30 CLS:WAD=10:DIMR$(3,30),RP(3):RP(1)=21
3:RP(2)=222:RP(3)=230:CP$(1)="O":CP$(2)=
"O":CP$(3)="@":CP$(4)="/":CP$(5)="-":CP$
(6)=" ":CP$(7)=CHR$(92):CP$(8)="O":CP$(9
)="O":CP$(10)="O"
40 'DRAW THE POKER MACHINE THEN READ IN
THE DATA FOR THE REELS
50 SCREEN0,,1:CLS:GOSUB510:FORR=1TO3:FOR
P=1TO30:READR$(R,P):IFLEN(R$(R,P))=1THEN
R$(R,P)="  "+R$(R,P):ELSER$(R,P)=" "+R$(
R,P)
60 NEXTP:NEXTR
70 LOCATE7,20:PRINT"IF YOU WANT TO TRY A
NOTHER GAME ANY TIME                PRESS <
RETURN>";:A$="":FORI=1TO1000:NEXT:LOCATE
7,18:PRINTCHR$(31);:LOCATE7,18:PRINT"TO
PUT IN COIN PRESS SPACE BAR";:MAX=30:R=F
NRN(RA):S=FNRN(RA):T=FNRN(RA)
80 BEEP:LOCATE17,7:PRINT"";:PRINTUSING"#
###";PAY;:IFWAD=0THEN630ELSELOCATE17,4:P
RINT"$";:PRINTUSING"###.##";WAD;:A$=INKE
Y$:IFA$=""THEN80ELSEIFA$=CHR$(13)THENCLS
:GOTO30ELSELOCATE17,18:PRINTCHR$(31);:WA
D=WAD-.2:LOCATE17,4:PRINT"$";
90 PRINTUSING"###.##";WAD;
100 'THIS LINE PULLS DOWN THE HANDLE,THE
 FOLLOWING LINES PRINTIN THE DIFFERENT P
OSITIONS
110 FORI=3TO5:PSET(83,I):NEXTI:FORI=3TO5
:PRESET(83,I):NEXTI:FORI=1TO14:ONIGOSUB1
60,160,160,150,150,150,140,140,140,130,1
40,150,160,170
120 NEXTI:GOTO190
130 FORD=1TO40:NEXTD:RETURN
140 LOCATE55,7:PRINT"   ";:LOCATE55,8:PRI
NT"   ";:LOCATE55,9:PRINT"   ";:LOCATE55,1
0:PRINTCHR$(135)+CHR$(135);:LOCATE55,11:
PRINTCHR$(138)+CHR$(32);:LOCATE55,12:PRI
NTCHR$(138)+CHR$(32);:RETURN
150 LOCATE55,7:PRINT"   ";:LOCATE55,8:PRI
NT"   ";:LOCATE55,9:PRINTCHR$(135)+CHR$(1
35);:LOCATE55,10:PRINTCHR$(138)+CHR$(32)
;:LOCATE55,11:PRINTCHR$(138)+CHR$(32);:R
ETURN
160 LOCATE55,7:PRINT"   ";:LOCATE55,8:PRI
NTCHR$(135)+CHR$(135);:LOCATE55,9:PRINTC
HR$(138)+CHR$(32);:LOCATE55,10:PRINTCHR$
(138)+CHR$(32);:RETURN
170 LOCATE55,7:PRINTCHR$(135)+CHR$(135);
:LOCATE55,8:PRINTCHR$(138)+CHR$(32);:LOC
ATE55,9:PRINTCHR$(138)+CHR$(32);:RETURN
180 '     DETERMINE APPROX. HOW MANY TIME
S EACH REEL TURNS OVER          SOMETIMES
THE REELS WILL TIP OVER JUST AFTER THEY
STOP
190 PAY=0:MAX=25:KK=FNRN(RA)+25:MAX=5:S1
=(KK+FNRN(RA)-3)*3/5:S2=(KK+FNRN(RA)-3)*
4/5:FORRS=1TOKK:MAX=20:IF RS>S2 AND FNRN
(RA)=1 THEN230ELSE IF RS>S2 THEN260ELSE
IF RS>S1 THEN230ELSE R=R-1:IF R<=1 THEN
R=30
200 REM PRINT CHARACTERS ON THE REELS
210 RA=R-1:RB=R-2:IFR=1THENRA=30:RB=29
220 LOCATE28,8:PRINTR$(1,R);:LOCATE28,7:
PRINTR$(1,RA);:LOCATE28,6:PRINTR$(1,RB);
230 T=T-1:IFT<1THENT=30
240 TA=T-1:TB=T-2:IFT=1THENTA=30:TB=29
250 LOCATE37,8:PRINTR$(2,T);:LOCATE37,7:
PRINTR$(2,TA);:LOCATE37,6:PRINTR$(2,TB);
260 S=S-1:IFS<1THENS=30
270 SA=S-1:SB=S-2:IFS=1THENSA=30:SB=29
280 LOCATE45,8:PRINTR$(3,S);:LOCATE45,7:
PRINTR$(3,SA);:LOCATE45,6:PRINTR$(3,SB);
290 NEXT
300 'CHECK ON PAYOUT,IF LUCKY, ADD TOGET
HER THE 3 CHRS ON REELS
310 TR$=R$(1,RA)+R$(2,TA)+R$(3,SA)
320 'CHECK IF JACKPOT
330 IFTR$="  A  A  A"THENLOCATE7,18:PRIN
T"JACKPOT - $50 -";:FORD=1TO100:NEXT:PAY
=50:LOCATE7,18:PRINTCHR$(131);:GOTO450
340 IFLEFT$(TR$,6)="  A  A"THENLOCATE7,1
8:PRINT"JACKPOT - $20 -";:FORD=1TO100:NE
XT:PAY=20:LOCATE7,18:PRINTCHR$(31);:GOTO
450
350 'CHECK IF RUN
360 IFTR$=" 10  J  Q"ORTR$="  Q  K  A"O
RTR$="  J  Q  K"THENPAY=4:GOTO440
370 X$=R$(1,RA):Y$=R$(2,TA):Z$=R$(3,SA)
380 '    CHECK IF A SMALLER PAYOUT
390 IFX$=Y$ANDY$=Z$ANDZ$=" 10"THENPAY=2:
GOTO440
400 IFX$=Y$ANDX$=" 10"THENPAY=1:GOTO440
410 IFX$=Y$ANDY$="  J"THENPAY=2:GOTO440
420 IFX$=" 10"THENPAY=.2:GOTO440
430 IFX$=Y$ANDY$=Z$THENPAY=6:GOTO440ELSE
80:'GOTO 40 IF NO PAYOUT
440 WAD=WAD+PAY:PAY=PAY*5:GOTO80
450 WAD=WAD+PAY:PAY=PAY*5:GOSUB510:GOTO8
0
460 'DATA FOR THE THREE REELS
470 DATA10,Q,K,A,J,Q,10,K,10,K,Q,K,J,10,
J,10,Q,K,K,J,10,10,Q,K,Q,K,J,K,Q,K
480 DATAK,Q,J,10,A,10,J,A,Q,K,Q,10,Q,K,Q
,J,10,10,J,10,K,10,J,Q,Q,10,K,10,10,K
490 DATA10,J,Q,10,A,K,Q,10,J,10,Q,K,A,10
,10,Q,K,K,A,Q,10,K,J,Q,10,J,Q,10,J,Q
500 'DRAW  THE POKER MACHINE ,PUT IN THE
 PAYOUT LISTING.
510 FORI=37TO87:PSET(I,18):PSET(I,8):PSE
T(I,0):PSET(I,42):PSET(I,37):NEXT:FORI=9
TO18:PSET(37,I):PSET(87,I):PSET(53,I):PS
ET(70,I):NEXT:FORI=1TO3:PSET(94,I+30):PS
ET(95,I+30):PSET(95,I+30):PSET(96,I+30):
NEXT:FORI=18TO32:PSET(97,I):PSET(98,I)
520 NEXT
530 LINE(200,45)-(400,75),PSET,B:LINE(18
0,75)-(420,130),PSET,B:LINE(220,0)-(380,
0),PSET:LINE(220,0)-(200,45),PSET:LINE(3
80,0)-(400,45),PSET:LINE(260,45)-(260,75
),PSET:LINE(340,45)-(340,75),PSET
540 GOSUB140:GOSUB150:GOSUB160:GOSUB170:
LINE(420,100)-(436,120),PSET,B
550 FORI=1TO3
560 PSET(96,14+I):PSET(99,14+I):PSET(96+
I,14):NEXT:FORI=8TO44:PSET(31,I):PSET(93
,I):NEXT:FORI=38TO41:PSET(36,I):PSET(88,
I):NEXT:FORI=32TO93:PSET(I,44):NEXT:PSET
(38,43):PSET(86,43):FORI=1TO7:PSET(30+I,
8-I):PSET(93-I,8-I):NEXT
570 FORI=2TO6:PSET(81,I):PSET(82,I):PSET
(84,I):PSET(85,I):NEXT:PSET(83,2):PSET(8
3,6)  :LOCATE33,1:PRINT"ARISTOCRAT";:LOCA
TE28,6:PRINT" J ";:LOCATE28,7:PRINT" Q "
;:LOCATE37,8:PRINT" K ";:LOCATE28,8:PRIN
T"10 ";:LOCATE37,7:PRINT" A";
580 LOCATE37,6:PRINT" Q ";:LOCATE45,7:PR
INT" K ";:LOCATE45,8:PRINT" J ";
590 LOCATE45,6:PRINT" Q";:FORI=11TO14STE
P3:PSET(38,I):PSET(52,I):PSET(54,I):PSET
(86,I):PSET(69,I):PSET(71,I):PSET(86,I):
NEXT:LOCATE18,3:PRINT"CREDIT";:LOCATE17,
4:PRINT"$";:PRINTUSING"###.##";WAD;:LOCA
TE25,10
```

```
600 PRINT"10  X  X :1    Q  Q  Q :30";:L
OCATE25,11:PRINT"10 10  X :5    K  K  K
:30";
610 LOCATE25,12:PRINT"10 10 10 :10   ANY
 RUN :20";:LOCATE25,13:PRINT" J  J  X :1
0   A  A  X $20";:LOCATE25,14:PRINT" J
J  J :30   A  A  A $50";:LOCATE7,7:PRINT
"COINS PAID";:RETURN
620 'IF YOU RUN OUT OF MONEY
630 CLS:LOCATE15,11:PRINT"YOU CANT PLAY
HERE WITH NO MONEY!":LOCATE15,11:PRINT"Y
OU HAVE JUST BEEN BOOTED OUT OF THE CLUB
!":END
```

****** LII/16K URANIUM CORE ******

TRS-80/SYSTEM-80

```
10 POKE16561,48:POKE16562,126:DEFINTA-Z:CLEAR200:CLS
20 REM ************************************
30 REM *     U R A N I U M   C O R E      *
40 REM *               (C)                *
50 REM *        BY PSIONIC SOFTWARE       *
60 REM ************************************
70 REM *     BY D.STEVENS + B.THOMAS      *
80 REM *  406 SANDGATE ROAD, SHORTLAND     *
90 REM *NEWCASTLE,N.S.W. 28/1/82-30/1/82*
100 REM************************************
110 FORL=0TO960STEP64:IFPEEK(L+15360)=32THENPRINT@L,"<<T>>RS-80
 OR  <<S>>YSTEM-80?";ELSENEXT
120 A$=INKEY$:IFA$="T"THEN130ELSEIFA$="S"THENOUT254,255ELSEGOTO1
20
130 FORK=1TO15:PRINT@L," ";:FORS=1TO5:PRINTCHR$(RND(63)+128);:NE
XT:PRINT"  U R A N I U M  C O R E  ";:FORS=1TO5:PRINTCHR$(RND(63
)+128);:NEXT:FORZ=1TO30:NEXT:L=L+1:NEXT:GOTO580
140 REM *** CONVERSION ***
150 PP=INT(X1)*2+INT(Y1)*64-1:RETURN
160 REM ** MOVE PLAYER **
170 P1=PEEK(C1):P2=PEEK(C2)
180 I$=INKEY$:IFI$<>" "ORGF=1THEN260
190 DL=10:Y1=PY:POKEC4,M:POKEC5,127:FORX1=PX+1TOPX+5:IFX1>30THEN
220
200 GOSUB150:IFPEEK(PP+C3)=174THENGF=1:A=USR(531):GOTO220
210 PRINT@PP,CHR$(140);:A=USR(5041):NEXT
220 IFGF=1THENDL=100
230 ZZ=X1:IFZZ>30THENZZ=30
240 FORX1=ZZTOPX+1STEP-1:GOSUB150:PRINT@PP,B$;:IFGF=1PRINT@PP-2,
UC$;ELSEPRINT@PP-2,CHR$(140);
250 A=USR(5041):NEXT:GF=1:CB=CB+1
260 IFP1=8THENVY=VY-2
270 IFP1=16THENVY=VY+2
280 IFP2=16THENVX=VX-2
290 IFP2=64THENVX=VX+2
300 IFBH=0THEN330
310 IFPX>CXTHENVX=VX-1ELSEIFPX<CXTHENVX=VX+1
320 IFPY>CYTHENVY=VY-1ELSEIFPY<CYTHENVY=VY+1
330 PRINT@937,CHR$(181);"X";CHR$(186);"       ";CHR$(181);"Y";CHR
$(186);"  ";:PRINT@951,VY;:PRINT@941,VX;
340 PRINT@2,SC;:FORL=1TO2:OX=PX:OY=PY:PX=PX+VX/6:PY=PY+VY/6:IFPY
<10RPY>13THENVY=VY*-1:IFPY<1THENPY=1ELSEPY=13
350 IFBH=0THEN390
360 IFPX<CX-20RPX>CX+20RPY<CY-20RPY>CY+2THEN390ELSEPX=INT(PX):PY
=INT(PY)
370 IFPX>CXTHENPX=PX-1ELSEIFPX<CXTHENPX=PX+1ELSEIFPY<CYTHENPY=PY
+1ELSEIFPY>CYTHENPY=PY-1ELSEIFPX=CXANDPY=CYTHEN560
380 X1=PX:Y1=PY:GOSUB150:PRINT@PP,P$;:FORL=1TO20:NEXT:GOTO370
390 IFPX<1THENX1=INT(OX):Y1=INT(OY):GOSUB150:PRINT@PP,B$;:GOTO49
0 REM ** ESCAPE **
400 IFPX>30THENPX=30:VX=VX*-1
410 X1=INT(PX):Y1=INT(PY):GOSUB150:IFPEEK(PP+C3)=174THENCB=CB+1
420 IFPEEK(PP+C3)=153THENPRINT@PP,P$;:X1=INT(OX):Y1=INT(OY):GOSU
B150:PRINT@PP,B$;:GOTO560 REM ***EXPLODES ***
430 X1=INT(OX):Y1=INT(OY):GOSUB150:PRINT@PP,B$;:X1=INT(PX):Y1=IN
T(PY):GOSUB150:PRINT@PP,P$;:POKEC4,M:POKEC5,127:A=USR(395)
440 NEXTL:PRINT@2,STRING$(6,176);:PRINT@30,HS(1);:X1=RND(30):Y1=
RND(13):GOSUB150:IFPEEK(PP+C3)=153THENQQ=PP+C3:FORK=1TO5:POKEQQ,
191:POKEQQ+1,191:POKEQQ,188:POKEQQ+1,188:POKEQQ,140:POKEQQ+1,140
:POKEQQ,128:POKEQQ+1,128:NEXTK
450 TT=TT+1:T=T+1:IFT>TLTHENT=0:Y1=RND(6)*2:X1=RND(12)*2+6:IFX1<
20ANDX1>12THENY1=Y1-1
460 IFTT=2THENTT=0:PRINT@FP," ";:FP=FP-1:IFFP=901THENGOTO560 REM
**NO FUEL EXPLODES ****
470 IFT=0THENGOSUB150:IFPEEK(PP+C3)<>174THENPRINT@PP,WB$;
480 GOTO170
490 IFCB=0THENPRINT@133,"MISSION UNSUCCESSFUL.";:FORL=1TO500:NEX
T:PRINT@197,"URANIUM CORE NOT PRESENT IN POD BAY.";:FORL=1TO500:
NEXT:PRINT@261,"YOU ARE RELIEVED OF YOUR COMMAND.";:FORL=1TO1500
:NEXT:GOTO860 REM **** HIGHSCORES ****
500 IFBH=1THEN X1=CX:Y1=CY:GOSUB150:POKEC4,M:POKEC5,127:FORL=1TO
5:PRINT@PP,BH$;:FORK=1TO20:NEXT:PRINT@PP,B$;:A=USR(2007):NEXTL
510 GF=0:SC=SC+LV*CB*5:LV=LV+1:VX=0:VY=0:PX=1:POKEC4,M:POKEC5,12
7:A=USR(444)
520 FP=CB*7+FP:IFFP>935THENFP=935
530 PRINT@897,"FUEL:";:POKEC4,M:POKEC5,127:FORL=902TOFP:PRINT@L,
CHR$(143);:A=USR(866):NEXT:CB=0:BH=0
540 GOTO790
550 REM*****EXPLODE*****
560 POKEC4,EX:POKEC5,126:POKE32524,PX*4-2:POKE32525,PY*3:A=USR(0
):POKE32524,PX*4-2:POKE32525,PY*3:A=USR(0)
570 FORK=1TOLEN(GO$):PRINT@470,LEFT$(GO$,K);:FORG=1TO70:NEXT:NEX
T:FORK=1TO500:NEXT:GOTO860
580 CLEAR2000:FORL=32717TO32767:READD:POKEL,D:NEXT:FORL=32688TO3
2716:READD:POKEL,D:NEXT:FORL=32305TO32651:READD:POKEL,D:NEXT
590 DATA1,0,4,33,0,60,126,254,32,40,17,254,92,242,238,127,11,62,
0,184,32,2,185,200,35,195,211,127,54,191,195,221,127,229,22,0,95
,33,63,1,237,82,125,225,119,195,221,127,255,255,255
```

```
600 DATA205,127,10,62,1,14,0,237,91,61,64,69,47,230,3,179,211,25
5,13,40,4,16,246,24,242,37,32,241,201
610 DATA33,0,60,54,128,17,1,60,1,255,3,237,176,201,62,198,24,6,6
2,134,24,2,62,70,50,124,126,122,6,255,4,214,3,242,79,126,198,3,2
03,39,79,104,38,0,6,6,41,16,253,22,0,203,59,48,1,12,25,17,0,60,2
5,203,33,203,33,203,33,58,124,126,129,50,124,126
620 DATA203,150,203,254,201,33,12,127,86,35,94,6,31,35,114,35,11
5,5,194,136,126,14,0,6,32,33,12,127,94,35,86,35,62,128,187,250,1
82,126,202,182,126,62,48,186,250,182,126,202,182,126,197,213,229
,205,63,126,225,209,193,5,194,151,126,65,62,1,211
630 DATA255,62,2,211,255,16,246,0,0,6,32,33,12,127,94,35,86,35,6
2,128,187,250,0,127,202,0,127,62,48,186,250,0,127,202,0,127,229,
213,197,205,67,126,193,33,76,127,22,0,88,25,25,209,126,131,95,35
,126,130,87,225,43,43,115,35,114,35,5,194,204,126
640 DATA12,62,14,185,194,146,126,201,62,167,77,92,60,88,84,48,10
4,40,118,26,104,12,90,240,34,240,20,12,6,26,34,40,51,48,73,48,90
,40,104,26,90,12,76,254,48,254,34,12,20,26,48,40,62,48,76,40,90,
26,76,12,62,254,48,12,34,26,62,40,76,26,62,12,255
650 DATA0,0,255,1,0,0,1,254,0,255,255,0,254,1,255,2,0,1,1,0,2,25
5,1,253,0,254,255,255,254,1,254,2,255,3,0,2,1,1,2,255,2,254,1,25
2,0,253,255,254,253,2,253,3,255,4,0,3,1,2,2,254,2,253,1
660 G$="PRESS <SPACEBAR> TO START GAME":R$="PRESS <ENTER> FOR IN
STRUCTIONS":FORL=1TO16:LB$=LB$+" "+CHR$(26)+CHR$(24):NEXT:UT$=CH
R$(181)+CHR$(186)+" "+CHR$(151)+CHR$(167)+" "+CHR$(159)+CHR$(175
)+" "+CHR$(151)+CHR$(175)+" "+CHR$(191)
670 UT$=UT$+" "+CHR$(181)+CHR$(186)+" "+CHR$(151)+CHR$(140)+CHR$
(171)+"   "+CHR$(183)+CHR$(179)+" "+CHR$(183)+CHR$(187)+" "+CHR$
(151)+CHR$(167)+" "+CHR$(189)+CHR$(176):GO$="G A M E - O V E R"
680 DIMHS$(10),HS(10):FORL=1TO15:VB$=VB$+CHR$(191)+CHR$(26)+CHR$
(24):NEXT:HB$=STRING$(62,131):TB$=STRING$(62,176):WB$=CHR$(153)+
CHR$(166):P$=CHR$(166)+CHR$(153):UC$=CHR$(174)+CHR$(157):BH$=CHR
$(183)+CHR$(187):B$=CHR$(128)+CHR$(128)
690 C1=14400:C2=14368:C3=15360:C4=16526:C5=16527:I=205:M=176:EX=
128:CS=49:POKE16527,127:POKEC4,M:FORK=360TO720STEP3:X=USR(K):NEX
T:GOTO860
700 REM ***** START GAME *****
710 LV=1:SC=0:SN=5:PX=1:PY=RND(13):XV=0:YV=0:T=0:TL=15:FP=935:GF
=0
720 POKEC4,CS:POKEC5,126:A=USR(0)
730 PRINT@0,TB$;:PRINT@960,HB$;:PRINT@0,VB$;:PRINT@61,VB$;
740 PRINT@897,"FUEL:";:POKEC4,M:POKEC5,127:FORL=902TOFP:PRINT@L,
CHR$(143);:A=USR(866):NEXT:FORL=1TO7:PRINT@937,STRING$(16,143);:
A=USR(970):PRINT@937,"VELOCITY SCANNER";:FORK=1TO70:NEXT:NEXTL
750 PRINT@453,"UNIVERSE";:PRINT@517,"INTERFACE";:FORL=514TO66STE
P-64:PRINT@L,"<==";:A=USR(5067):PRINT@L,"   ";:NEXT:FORL=66TO834
STEP64:PRINT@L,"<==";:A=USR(5067):PRINT@L,"   ";:NEXT
760 FORL=834TO514STEP-64:PRINT@L,"<==";:A=USR(5067):PRINT@L,"
";:NEXT
770 PRINT@453,CHR$(200);:PRINT@517,CHR$(201);:X1=PX:Y1=PY:GOSUB1
50:PRINT@PP,P$;:A=USR(531)
780 REM *** NEW CORE ***
790 CX=RND(10)+15:CY=RND(11)+1:X1=CX:Y1=CY:GOSUB150:POKEC4,M:POK
EC5,127:FORL=1TO7:PRINT@PP,B$;:FORK=1TO50:NEXT:PRINT@PP,UC$;:FOR
K=1TO50:NEXT:A=USR(1000):NEXTL
800 X1=CX-1:Y1=CY:GOSUB150:PRINT@PP,UC$;:X1=CX+1:GOSUB150:PRINT@
PP,UC$;:Y1=CY+1:X1=CX:GOSUB150:PRINT@PP,UC$;:Y1=CY-1:GOSUB150:PR
INT@PP,UC$;
810 IFLV<4THEN820ELSEX1=CX-1:Y1=CY-1:GOSUB150:IFPEEK(PP+C3)=174T
HENBH=1ELSEX1=CX+1:GOSUB150:IFPEEK(PP+C3)=174THENBH=1ELSEY1=CY+1
:GOSUB150:IFPEEK(PP+C3)=174THENBH=1ELSEX1=CX-1:GOSUB150:IFPEEK(P
P+C3)=174THENBH=1
820 IFBH=0THEN840ELSEFORX1=CX-2TOCX+2:FORY1=CY-2TOCY+2:IFX1<1ORX
1>30ORY1<1ORY1>13THEN830ELSEGOSUB150:PRINT@PP,B$;
830 NEXT:NEXT:X1=CX:Y1=CY:GOSUB150:PRINT@PP,BH$;
840 GOTO170
850 REM *** HIGH SCORES ***
860 GF=0:VX=0:VY=0:BH=0:CB=0:LV=1:Z=0:FORL=10TO1STEP-1:IFSC>HS(L
)THENZ=L
870 NEXTL:IFZ=0THENZ=1:GOTO970:ELSECLS:PRINT@0,TB$;:PRINT@960,HB
$;:PRINT@0,VB$;:PRINT@61,VB$;: PRINT@77,UT$;
880 PRINT@205,"** C O N G R A T U L A T I O N S **";:PRINT@327,"
THE INTERSTELLAR HIGH COMMAND WISHES TO INFORM";:PRINT@386,"YOU
THAT YOU HAVE ONE OF THE BEST MISSION RECORDS TO DATE.";:PRINT@5
26,"PLEASE TYPE IN YOUR IDENTITY-CODE";
890 N$="":I$=INKEY$:PRINT@666,CHR$(188);STRING$(6,140);CHR$(188)
;:PRINT@730,CHR$(191);:PRINT@737,CHR$(191);:PRINT@794,STRING$(8,
131);:POKEC4,M:POKEC5,127:FORL=700TO600STEP-5:X=USR(L+500):FORK=
1TO50:NEXT:NEXT
900 I$=INKEY$:IFI$=CHR$(8)THENIFLEN(N$)=0GOTO900ELSEN$=LEFT$(N$,
(LEN(N$)-1)):PRINT@732,"     ";:GOTO930
910 IFI$=CHR$(10)THEN900
920 IFI$=CHR$(13)THEN950ELSEN$=N$+I$
930 IFLEN(N$)>4THENN$=LEFT$(N$,4)
940 PRINT@732,N$;:GOTO900
950 POKEC5,127:POKEC4,M:FORK=1TO30:A=USR(RND(255)+270):NEXT:N$=N
$+"     ":N$=LEFT$(N$,4):IFN$="     "THENN$="NONE"
960 FORL=10TO(Z+1)STEP-1:HS$(L)=HS$(L-1):HS(L)=HS(L-1):NEXT:HS$(
Z)=N$:HS(Z)=SC
970 CLS:PRINT@0,STRING$(64,131);:PRINT@960,STRING$(63,176);:FORL
=0TO896STEP64:PRINT@L,CHR$(191);:PRINT@L+63,CHR$(191);:NEXT:POKE
16320,191:POKE16383,191:PRINT@49,CHR$(139);"HIGH SCORES";CHR$(13
5);:PS=115
980 FORL=1TO10:IFHS(L)=0THEN990ELSEPRINT@PS,HS$(L);HS(L);
990 PS=PS+64:NEXTL:PS=(Z-1)*64+115:IFPS<115THENPS=115
1000 REM *** TITLE ***
1010 PRINT@755,CHR$(139);STRING$(6,191);CHR$(135);:PRINT@820,CHR
$(130);STRING$(4,143);CHR$(129);:PRINT@879,CHR$(136);STRING$(5,1
91);CHR$(189);CHR$(146);CHR$(161);CHR$(190);STRING$(5,191);CHR$(
132);
1020 PRINT@944,CHR$(130);CHR$(143);STRING$(2,191);CHR$(159);CHR$
(135);"  ";CHR$(131);CHR$(143);STRING$(2,191);CHR$(159);CHR$(129
);:PRINT@135,CHR$(191);:PRINT@140,CHR$(191);:PRINT@199,CHR$(191)
;:PRINT@204,CHR$(191);:PRINT@263,CHR$(191);:PRINT@268,CHR$(191);
1030 PRINT@327,CHR$(143);STRING$(4,176);CHR$(143);:PRINT@205,CHR
$(160);STRING$(2,176);"  ";STRING$(2,176);"     ";CHR$(176);CHR$
(160);CHR$(144);CHR$(176);"   ";CHR$(176);"     ";CHR$(176);
1040 PRINT@269,CHR$(177);CHR$(140);CHR$(142);" ";CHR$(191);"  ";
CHR$(191);CHR$(140);CHR$(176);"  ";CHR$(191);CHR$(170);CHR$(149)
;CHR$(191);"   ";CHR$(191);CHR$(131);CHR$(140);CHR$(176);CHR$(14
0);CHR$(131);CHR$(191);
```

```
1050 PRINT@334,CHR$(131);CHR$(140);CHR$(176);CHR$(191);STRING$(2
,131);CHR$(191);"  ";CHR$(131);CHR$(140);CHR$(191);CHR$(170);CHR
$(149);CHR$(143);STRING$(3,176);CHR$(143);CHR$(197);CHR$(191);
1060 PRINT@460,CHR$(176);STRING$(2,140);STRING$(2,131);CHR$(137)
;CHR$(132);:PRINT@523,CHR$(190);CHR$(129);CHR$(199);STRING$(4,17
6);" ";STRING$(3,176);CHR$(144);" ";STRING$(5,176);
1070 PRINT@587,CHR$(175);CHR$(144);CHR$(198);CHR$(191);"    ";CH
R$(191);CHR$(176);CHR$(152);CHR$(140);CHR$(134);" ";CHR$(191);ST
RING$(2,176);
1080 PRINT@652,CHR$(131);STRING$(2,140);STRING$(3,176);CHR$(140)
;CHR$(139);STRING$(4,176);CHR$(143);" ";CHR$(130);CHR$(140);STRI
NG$(2,176);CHR$(191);STRING$(4,176);
1090 FORL=1TO50:I$=INKEY$:IFI$=" "THEN710ELSEIFI$=CHR$(13)THEN12
10ELSENEXTL
1100 FORX=422TO678STEP64:PRINT@X,CHR$(191);:NEXT:PRINT@742,CHR$(
143);:FORX=741TO708STEP-1:PRINT@X,CHR$(140);:NEXT:PRINT@708,CHR$
(143);:FORX=644TO132STEP-64:PRINT@X,CHR$(191);:NEXT:PRINT@68,CHR
$(188);:FORX=69TO107:PRINT@X,CHR$(140);:NEXT:PRINT@108,CHR$(188)
;
1110 I$=INKEY$:IFI$=" "THEN710ELSEIFI$=CHR$(13)THEN1210
1120 FORX=172TO748STEP64:PRINT@X,CHR$(191);:NEXT:FORX=812TO770ST
EP-1:PRINT@X,CHR$(131);:NEXT:FORX=706TO66STEP-64:PRINT@X,CHR$(19
1);:NEXT:FORX=2TO46:PRINT@X,CHR$(179);:NEXT:FORX=110TO814STEP64:
PRINT@X,CHR$(191);:NEXT:PRINT@813,CHR$(176);
1130 I$=INKEY$:IFI$=" "THEN710ELSEIFI$=CHR$(13)THEN1210
1140 POKEC5,127:FORX=812TO770STEP-1:PRINT@X,CHR$(179);:NEXT:ZZ=1
00:FORL=1TO10:POKEC4,I:A=USR(0):POKEC4,M:A=USR(290):FORK=1TOZZ:N
EXT:ZZ=ZZ-10:NEXTL:POKEC4,I
1150 FORL=1TO10:A=USR(0):POKEC4,M:A=USR(290):POKEC4,I:IFINKEY$="
 "THEN710:ELSEIFINKEY$=CHR$(13)THEN1210ELSENEXTL
1160 PRINT@478,UC$;:ID=1:GD=-1:Q=906:Q1=842:POKEC4,M:FORL=1TO300
:Q=Q+GD:IFQ>911ORQ<899THENGD=-GD
1170 Q1=Q1+ID:IFQ1<835ORQ1>846THENID=-ID
1180 A=USR(304):PRINT@Q," ";G$;" ";:PRINT@Q1," ";R$;" ";:PRINT@P
S,HS$(Z);:FORK=1TO20:NEXT
1190 PRINT@Q+8,CHR$(200);:PRINT@Q1+8,CHR$(197);:PRINT@PS,CHR$(19
6);:FORK=1TO20:NEXT
1200 I$=INKEY$:IFI$=CHR$(13)THEN1210ELSEIFI$=" "THEN710ELSENEXTL
1210 POKE32524,60:POKE32525,22:POKEC4,EX:POKEC5,126:A=USR(0):POK
E32524,60:POKE32525,22:A=USR(0):POKEC4,M:POKEC5,127
1220 FORL=0TO62STEP2:A=USR(1000):PRINT@L,LB$;:PRINT@L+1,LB$;:NEX
T:CLS:PRINT@960,STRING$(63,176);:PRINT@0,STRING$(64,131);:FORL=0
TO896STEP64:PRINT@L,CHR$(191);:PRINT@L+63,CHR$(191);:NEXT:POKE16
320,191:POKE16383,191:POKEC4,M
1230 PRINT@77,UT$;:PRINT@197,P$;"   VECTORED PROBABILITY SHIFT P
OD.";:PRINT@268,"PILOT THIS TO RETRIEVE URANIUM  CORES.";:PRINT@
325,UC$;"   URANIUM CORE.";:PRINT@396,"RETURN THESE TO UNIVERSE
INTERFACE TO REFUEL.";
1240 PRINT@453,WB$;"   UNSTABLE NEUTRINO WEB.";:PRINT@524,"DANGE
ROUS. THESE EXPLODE ON CONTACT. ***********";:PRINT@581,BH$;"
BLACK HOLE.";CHR$(220);"*  AVOID  *";:PRINT@652,"HI-GRAVITY WARP
S SHIFT DRIVE OF POD. ***********";
1250 PRINT@714,"THRUST CONTROLS -->";CHR$(199);CHR$(183);"[";CHR
$(187);" UP.   ";CHR$(183);CHR$(92);CHR$(187);" DOWN.";:PRINT@79
9,CHR$(183);"<";CHR$(187);" LEFt. ";CHR$(183);">";CHR$(187);" RI
GHT.";
1260 PRINT@837,STRING$(6,191);CHR$(183);"SPACE";CHR$(187);STRING
$(6,191);" PROBABILITY GRAPPLE (RANGE 5).";:PRINT@921,"ONE PER F
RAME ONLY.";:PRINT@977,"PRESS <SPACEBAR> TO START GAME";
1270 POKEC4,M:FORV=1TO300:I$=INKEY$:IFI$=" "THENPRINT@1023,"";:F
ORL=1TO16:PRINT:A=USR(L+300):NEXT:GOTO710
1280 PRINT@984,CHR$(200);:PRINT@628,"AVOID";:FORJ=1TO100:NEXT:PR
INT@984,"SPACEBAR";:PRINT@628,CHR$(197);:FORJ=1TO100:NEXT:FORK=3
00TO304:X=USR(K):NEXT:NEXT
1290 POKEC4,EX:POKEC5,126:FORK=1TO15:POKE32524,RND(127):POKE3252
5,RND(47):A=USR(0):NEXT:GOTO970
```

****** SINGLE KEY MENU ******

MODEL III

```
00010 ;*****************************************************
00020 ;******       MENU.....A COMMAND MENU        **********
00030 ;******      FOR NEWDOS/80 V2.0 MOD III      **********
00040 ;******        (c) 1982 by S.J.TURTLE        **********
00050 ;******          HILL END, BRISBANE          **********
00060 ;*****************************************************
00070          ORG     0F200H
00080 START    LD      A,0EH
00090          CALL    0033H           ; TURN ON THE CURSOR
00100          CALL    0049H           ; GET SOME INPUT
00110          CP      0DH             ; IS IT <ENTER> ?
00120          JR      Z,MENU          ; YES? THEN GO TO MENU
00130          PUSH    HL              ; NO? THEN GO GET MORE
00140          LD      C,B             ; FROM ROM ROUTINE.
00150          JP      05E3H
00160 MENU     LD      A,0FH
00170          CALL    0033H           ; TURN OFF CURSOR
00180          CALL    01C9H           ; CLEAR THE SCREEN
00190          LD      HL,3D00H
00200          LD      (4020H),HL
00210          LD      HL,TITLE
00220          CALL    021BH           ; PRINT THE TITLE
00230          LD      HL,3C1AH
00240          LD      (4020H),HL
00250          LD      HL,0F2F0H
00260          LD      B,00
00270 LOOP     LD      A,(HL)          ; PRINT THE MENU ITSELF
00280          CP      0FFH            ; IF FFhex THEN NO MORE
00290          JR      Z,DOMENU
00300          CP      00              ; IF 00hex THEN NEXT LINE
00310          JR      NZ,PRINT        ; OTHERWISE PRINT CHAR.
00320          PUSH    HL              ; SAVE BUFFER ADDRESS
00330          LD      HL,(4020H)      ; GET THE CURSOR POSITION
00340          LD      (HL),A          ; AND WRITE 00 TO SCREEN
00350          EX      DE,HL           ; DE = CURSOR POSITION
00360          POP     HL              ; GET BUFFER ADDR BACK
00370          JR      NEXTLN          ; AND DO ANOTHER ONE
00380 PRINT    CALL    0033H           ; PRINT THE CHARACTER
00390          INC     B               ; ADD ONE TO COUNTER
```

```
00400           INC     HL              ; INCREM. BUFFER POINTER
00410           JR      LOOP            ; GET NEXT CHARACTER
00420 NEXTLN    INC     HL              ; INCREM. BUFFER POINTER
00430           PUSH    HL              ; AND SAVE IT.
00440           PUSH    DE              ; SAVE LAST CURSOR POS.
00450           LD      HL,(4020H)      ; GET CURRENT CURSOR POS.
00460           LD      DE,40H
00470           ADD     HL,DE           ; MOVE DOWN 1 LINE
00480           LD      A,L
00490           SUB     B               ; MOVE BACK THIS MANY
00500           LD      L,A
00510           LD      (4020H),HL      ; AND RESET CURSOR POSN.
00520           LD      B,00            ; RESET COUNTER
00530           POP     DE              ; RESTORE REGISTERS
00540           POP     HL
00550           JR      LOOP            ; DO NEXT LINE
00560 DOMENU    LD      (LASTLN),DE     ; SAVE LAST LINE FOR REF.
00570           LD      HL,3C16H        ; = TOP OF MENU
00580           PUSH    HL
00590           CALL    DRAW            ; DRAW POINTER ON SCREEN
00600 KEYIN     CALL    0049H           ; WAIT FOR INPUT
00610           CP      0DH             ; <ENTER> ?
00620           JR      Z,DOIT          ; THEN DO MENU COMMAND
00630           CP      0AH             ; DOWN ARROW ?
00640           JR      Z,DOWN          ; THEN MOVE DOWN ONE
00650           CP      5BH             ; UP ARROW ?
00660           JR      Z,UP            ; THEN MOVE UP ONE
00670           CP      1FH             ; <CLEAR> ?
00680           JR      NZ,KEYIN        ; NO? THEN TRY AGAIN
00690           CALL    01C9H           ; OTHERWISE CLEAR SCREEN
00700           LD      HL,51CCH        ; PRINT NEWDOS READY
00710           CALL    021BH
00720           POP     DE              ; REORGANIZE STACK
00730           LD      HL,4225H        ; SET UP BUFFER POINTER
00740           LD      B,04FH          ; NO OF CHARACTERS
00750           JP      0040H           ; JUMP TO ROM LINE INPUT
00760 ;
00770 DOWN      POP     HL
00780           CALL    BLANK           ; ERASE POINTER
00790           LD      DE,60           ; MOVE DOWN ONE LINE
00800           ADD     HL,DE
00810           LD      DE,(LASTLN)
00820           RST     18H             ; IS IT TOO FAR DOWN
00830           JR      C,$+11          ; NO?
00840           JR      NC,$+6          ; YES?
00850           BIT     6,H             ; PAST END OF SCREEN?
00860           JR      Z,$+5           ; NO?
00870           LD      HL,3C16H        ; TOO FAR THEN BACK TO TOP
00880           JR      DOMENU+7        ; GO AND GET ANOTHER ONE
00890 ;
00900 UP        POP     HL
00910           CALL    BLANK           ; ERASE POINTER
00920           LD      DE,-68          ; MOVE UP ONE LINE
00930           ADD     HL,DE
00940           BIT     2,H             ; TOO HIGH?
00950           JR      NZ,$+5          ; NO?
00960           LD      HL,3C16H        ; IF YES THEN STAY THERE
00970           JR      DOMENU+7        ; GO TRY ANOTHER ONE
00980 BLANK     XOR     A               ; A=0
00990           LD      B,4
01000 LOOP1     LD      (HL),A          ; WRITE 4 SPACES
01010           INC     HL
01020           DJNZ    LOOP1
01030           RET                     ; THATS ALL!
01040 DRAW      LD      DE,POINTR
01050           EX      DE,HL
01060           LD      BC,04
01070           LDIR                    ; BLOCK MOVE POINTER
01080           RET
01090 ;
01100 DOIT      POP     HL              ; GET CURSOR POSITION
01110           LD      DE,4225H        ; = INPUT BUFFER ADDRESS
01120           LD      A,4
01130           ADD     A,L
01140           LD      L,A             ; POINT HL TO COMMAND STR.
01150 LOOP3     LD      A,(HL)          ; GET CHARACTER
01160           CP      00              ; IS IT 00 ?
01170           JR      Z,PROC          ; YES ? MUST BE END
01180           LD      (DE),A          ; MOVE TO BUFFER
01190           INC     DE              ; NEXT ONE
01200           INC     HL              ; NEXT CHARACTER
01210           JR      LOOP3           ; GET IT
01220 PROC      LD      A,0DH           ; PUT CARRIAGE RETURN
01230           LD      (DE),A          ; INTO INPUT BUFFER
01240           CALL    01C9H           ; CLEAR SCREEN
01250           LD      HL,4225H        ; HL = INPUT BUFFER
01260           RET                     ; "RETURN" TO DOS AT 497B
01270 ;
01280 POINTR    DEFB    244             ; POINTER CHARACTERS
01290           DEFB    245
01300           DEFB    246
01310           DEFB    32              ; + ONE SPACE
01320           DEFB    03              ; AND END OF TEXT
01330 ;
01340 ;
01350           ORG     0F2F0H          ; START OF COMMAND BUFFER
01360           DEFM    'BASIC2/CMD'    ; ALL COMMANDS ARE
01370           DEFB    00              ; ENTERED AS A DEFM
01380           DEFM    'DIR'           ; STATEMENT WITH THE
01390           DEFB    00              ; COMMAND....FOLLOWED BY
01400           DEFM    'HIMEM'         ; A DEFB STATEMENT OF '00'
01410           DEFB    00              ; ANY NUMBER UP TO AND
01420           DEFM    'FREE'          ; INCLUDING 16 CAN BE
01430           DEFB    00              ; ENTERED.
01440           DEFM    'LIB'
01450           DEFB    00
01460           DEFM    'CLEAR MEM=0EFFEH'      ; IF A CLEAR
01470           DEFB    00              ; STATEMENT IS USED IT
01480           DEFM    'MDRET'         ; MUST SET MEM AT F1FEhex
01490           DEFB    00              ; OTHERWISE MENU/CMD WILL
01500           DEFM    'SYSTEM 0'      ; BE ERASED.
01510           DEFB    00
```

```
01520           DEFM    'PDRIVE 0'
01530           DEFB    00
01540           DEFM    'ROUTE'
01550           DEFB    00
01560           DEFM    'SETCOM'
01570           DEFB    00
01580           DEFM    'FORMS'
01590           DEFB    00
01600           DEFM    'CLOCK'
01610           DEFB    00
01620           DEFM    'CLOCK,N'
01630           DEFB    00
01640           DEFM    'TIME'
01650           DEFB    00
01660           DEFM    'DATE'
01670           DEFB    00              ; THE LAST COMMAND MUST
01680           DEFB    0FFH            ; BE FOLLOWED BY FFhex.
01690 LASTLN    DEFW    3FFFH           ; THIS IS THE LAST LINE
01700 ;         POINTER AND WILL BE AUTOMATICALLY UPDATED
01710 ;         BY THE PROGRAM.
01720 TITLE     DEFM    'NEWDOS/80 Ver 2.0'     ; THE TITLE BLOCK
01730           DEFB    1DH             ; EACH LINE SEPARATED BY
01740           DEFB    0AH             ; A CARRIAGE RETURN AND
01750           DEFM    '##################'     ; LINE FEED.
01760           DEFB    1DH
01770           DEFB    0AH
01780           DEFM    '       MENU        '
01790           DEFB    1DH
01800           DEFB    0AH
01810           DEFM    '       ####        '
01820           DEFB    1DH
01830           DEFB    0AH
01840           DEFB    0AH
01850           DEFM    'Use arrow keys to'
01860           DEFB    1DH
01870           DEFB    0AH
01880           DEFM    'move pointer then'
01890           DEFB    1DH
01900           DEFB    0AH
01910           DEFM    '  press <ENTER>.   '
01920           DEFB    1DH
01930           DEFB    0AH
01940           DEFM    'Press <CLEAR> to'
01950           DEFB    1DH
01960           DEFB    0AH
01970           DEFM    '   enter text.   '
01980           DEFB    03H
01990           END     402DH           ; ENTRY POINT OF DOS.
```

```
F200:  3E 0E CD 33 00 CD 49 00 FE 0D 28 05 E5 48 C3 E3
F210:  05 3E 0F CD 33 00 CD C9 01 21 00 3D 22 20 40 21
F220:  65 F3 CD 1B 02 21 1A 3C 22 20 40 21 F0 F2 06 00
F230:  7E FE FF 28 2A FE 00 20 09 E5 2A 20 40 77 EB E1
F240:  18 07 CD 33 00 04 23 18 E7 23 E5 D5 2A 20 40 11
F250:  40 00 19 7D 90 6F 22 20 40 06 00 D1 E1 18 D1 ED
```

```
F260:  53 63 F3 21 16 3C E5 CD C2 F2 CD 49 00 FE 0D 28
F270:  5B FE 0A 28 1A FE 5B 28 30 FE 1F 20 ED CD C9 01
F280:  21 CC 51 CD 1B 02 D1 21 25 42 06 4F C3 40 00 E1
F290:  CD BA F2 11 3C 00 19 ED 5B 63 F3 DF 38 09 30 04
F2A0:  CB 74 28 03 21 16 3C 18 BD E1 CD BA F2 11 BC FF
F2B0:  19 CB 54 20 03 21 16 3C 18 AC AF 06 04 77 23 10
F2C0:  FC C9 11 E8 F2 EB 01 04 00 ED B0 C9 E1 11 25 42
F2D0:  3E 04 85 6F 7E FE 00 28 05 12 13 23 18 F6 3E 0D
F2E0:  12 CD C9 01 21 25 42 C9 F4 F5 F6 20 03 2B EE CD
F2F0:  42 41 53 49 43 32 2F 43 4D 44 00 44 49 52 00 48
F300:  49 4D 45 4D 00 46 52 45 45 00 4C 49 42 00 43 4C
F310:  45 41 52 20 4D 45 4D 3D 30 45 46 46 45 48 00 4D
F320:  44 52 45 54 00 53 59 53 54 45 4D 20 30 00 50 44
F330:  52 49 56 45 20 30 00 52 4F 55 54 45 00 53 45 54
F340:  43 4F 4D 00 46 4F 52 4D 53 00 43 4C 4F 43 4B 00
F350:  43 4C 4F 43 4B 2C 4E 00 54 49 4D 45 00 44 41 54
F360:  45 00 FF FF 3F 4E 45 57 44 4F 53 2F 38 30 20 56
F370:  65 72 20 32 2E 30 1D 0A 23 23 23 23 23 23 23 23
F380:  23 23 23 23 23 23 23 23 23 1D 0A 20 20 20 20 20
F390:  20 4D 45 4E 55 20 20 20 20 20 20 20 1D 0A 20 20
F3A0:  20 20 20 20 23 23 23 23 20 20 20 20 20 20 20 1D
F3B0:  0A 0A 55 73 65 20 61 72 72 6F 77 20 6B 65 79 73
F3C0:  20 74 6F 1D 0A 6D 6F 76 65 20 70 6F 69 6E 74 65
F3D0:  72 20 74 68 65 6E 1D 0A 20 20 70 72 65 73 73 20
F3E0:  3C 45 4E 54 45 52 3E 2E 20 20 1D 0A 50 72 65 73
F3F0:  73 20 3C 43 4C 45 41 52 3E 20 74 6F 1D 0A 20 20
F400:  20 65 6E 74 65 72 20 74 65 78 74 2E 20 20 03
```

****** LII/4K LEVEL 2 DEFUSR FUNCTION ******

TRS-80/SYSTEM-80

```
00100 ;* = * = * = * = * = * = * = * = * = * = * = * = * = * = *
00110 ;*              DEFUSR PROCESSOR
00120 ;*      (C)COPYRIGHT ROGER BOWLER 1982
00130 ;* = * = * = * = * = * = * = * = * = * = * = * = * = * = *
00140           ORG     415BH
00150           JP      DEFPRC
00160           ORG     4040H
00170 DEFPRC    RST     8       ;?SN ERROR IF NOT "DEFUSR"
00180           DEFB    0C1H    ;CODE FOR "USR"
00190           RST     8       ;CHECK THAT "=" FOLLOWS
00200           DEFB    0D5H    ;CODE FOR "="
00205           CALL    2337H   ;EVALUATE EXPRESSION
00210           PUSH    HL      ;SAVE TEXT POINTER
00220           CALL    0A7FH   ;LOAD EXPRESSION VALUE INTO HL
00230           LD      (408EH),HL;SET USR ENTRY VECTOR
00240           POP     HL      ;RESTORE TEXT POINTER
00250           RET             ;AND RETURN
00260           END     06CCH
```

```
4040:  CF C1 CF D5 CD 37 23 E5 CD 7F 0A 22 8E 40 E1 C9
```

```
10 'DEFUSR PROGRAM 2 (c)Copyright Roger Bowler 1982
20 FOR I=16448 TO 16463 'HEX 4040 to 404F
30 READ J: POKE I,J: NEXT I
40 DATA 207,193,207,213,205,055,035,229
50 DATA 205,127,010,034,142,064,225,201
60 POKE 16732,64: POKE 16733,64 'HEX 415C/D


10 'DEFUSR PROGRAM 3 (DEFUSR DEMONSTRATION)
20 FOR X=32000 TO 32013
30 READ A: POKE X,A
40 NEXT X
50 DATA 33,0,60,54,191,17,1,60,1,255,3,237,176,201
60 DEFUSR=32000
70 X=USR(0) 'WHITE-OUT SCREEN
80 FOR X=1 TO 1000: NEXT X 'DELAY LOOP
```

****** L1/4K COMPOUND MULTIPLICATION & LONG DIVISION ******

TRS-80/SYSTEM-80

```
5  REM C.STOBERT  03-795 6590
10  CLS
20  P.:P."COMPOUND MULTIPLICATION AND LONG DIVISION":P.
30  P.:P."OCCASIONALLY THE (?) WILL ASK FOR"
40  P."AN ENTRY WHICH IS ZERO."
50  P."WHEN THIS OCCURS ENTER THE 0."
100  P.:P."ENTER <1> IF YOU WANT MULTIPLICATION PROBLEMS"
110  IN."AND <2> IF LONG DIVISION ";P
115  IF(P<>1)*(P<>2) T.CLS:G.100
120  P.:IN."ENTER YOUR NAME PLEASE ";B$
130  R=0:T=0:0=0
200  CLS:P.A.0,"";:IFT=0P."O.K. ";:G.280
210  Q=RND(6):ONQG.220,230,240,250,260,270
220  P."KEEP IT UP ";:G.280
230  P."GOOD GOING ";:G.280
240  P."GOOD WORK ";:G.280
250  P."TOP EFFORT ";:G.280
260  P."NICE PROGRESS ";:G.280
270  P."NOT TOO BAD IS IT ";
280  P.B$:P.A.68,"NOW- ";
290  Q=RND(6):ONQG.300,310,320,330,340,350
300  P."HAVE A TRY AT THIS":G.360
310  P."NEXT PROBLEM":G.360
320  P."MIND BENDER #";R+1:G.360
330  P."TRY ANOTHER ONE":G.360
340  P."SEE IF THIS STOPS YOU":G.360
350  P."HEAD SHRINKER #";R+1
360  IFP=2T.1400
400  IFO=1T.420
410  A=RND(149)+100:B=RND(299)+100
420  C=INT(A/100):D=INT((A-C*100)/10)
430  E=A-C*100-D*10
440  F=INT(B/100):G=INT((B-F*100)/10)
450  H=B-F*100-G*10:T=T+1:READX,Y,Z
470  P.A.X,C:P.A.Y,D:P.A.Z,E:READX,Y,Z
490  P.A.X,F:P.A.Y,G:P.A.Z,H;:P."  X"
500  F.N=415TO419:GOS.1595:N.N
510  F.N=667TO675:GOS.1595:N.N
520  F.N=795TO803:P.A.N,"=":N.N
550  N=1:L=0
560  IF(N=1)+(N=2)T.READV,W,X,Y
570  IFN=3T.READV,W,X
580  IFN=4T.READV,W,X,Y,Z
590  P.A.V,:IN.C:GOS.990
600  P.A.W,:IN.D:GOS.980
610  P.A.X,:IN.E:GOS.970
620  IF(N=1)*(H*A>999)T.P.A.Y,:IN.K:GOS.960
630  IF(N=2)*(G*A>999)T.P.A.Y,:IN.K:GOS.960
640  IFN<4T.N=N+1:G.560
650  P.A.Y,:IN.K:GOS.960
660  IFA*B>9999T.P.A.Z,:IN.L:GOS.950
670  M=C+10*D+100*E+1E3*K+1E4*L
680  IFA*B=MT.P.A.832,"WELL DONE!":O=2:R=R+1:G.700
690  P.A.832,"SORRY THERE'S A MISTAKE SOMEWHERE":O=1
700  P.:IN."PRESS <ENTER> TO CONTINUE ";A$
710  IFR=10T.800
720  REST.:CLS:G.200
800  CLS:P.:P."THAT IS 10 PROBLEMS ";B$
810  P.:P."YOU HAD";T;"TRIES"
820  P."WHICH GAVE YOU A MARK OF";(1-(T-10)/10)*100;"%"
840  P.:P.:IN."PRESS <ENTER> TO CONTINUE ";A$:REST.:CLS:G.100
900  D.286,288,290,350,352,354
910  D.482,480,478,476
920  D.544,542,540,538
930  D.606,604,602
940  D.738,736,734,732,730
950  P.A.Z,L
960  P.A.Y,K
970  P.A.X,E
980  P.A.W,D
990  P.A.V,C:RET.
1300  READ V,W,X
1310  P.A.V,:IN.C:GOS.990
1320  P.A.W,:IN.D:GOS.980
1330  P.A.X,:IN.E:GOS.970
1340  RET.
1400  F.N=1TO22:READZ:N.N:IFO=1T.1420
1410  A=RND(299)+100:B=RND(8999)*RND(10)+10000
1420  F=INT(B*1E-4):G=INT((B-F*1E4)*1E-3)
1430  H=INT((B-F*1E4-G*1E3)/100)
1440  I=INT((B-F*1E4-G*1E3-H*100)/10)
1450  J=B-F*1E4-G*1E3-H*100-I*10
1460  C=INT(A/100):D=INT((A-C*100)/10)
1465  E=A-100*C-10*D:T=T+1
1470  READ X,Y,Z
1480  P.A.X,C:P.A.Y,D:P.A.Z,E
1580  READV,W,X,Y,Z
```

```
1590  P.A.V,F:P.A.W,G:P.A.X,H:P.A.Y,I:P.A.Z,J:G.1600
1595  P.A.N,"-":RET.
1600  F.N=46TO65:S.(N,10):N.N
1610  F.Y=10TO14:S.(45,Y):N.Y
1620  F.N=408TO412:GOS.1595:N.N
1630  F.N=600TO606:GOS.1595:N.N
1640  F.N=794TO800:GOS.1595:N.N
1650  F.N=922TO928:P.A.N,"=":N.N:K=0:READZ
1680  P.A.Z,:IN.G:P.A.Z,G
1690  GOS.1300:GOS.1300
1700  READZ:P.A.Z,I:READZ
1720  P.A.Z,:IN.H:P.A.Z,H:GOS.1300
1740  READY:IFH*A>999T.P.A.Y,:IN.K:GOS.960
1750  GOS.1300:READZ:P.A.Z,J
1760  READZ:P.A.Z,:IN.I:P.A.Z,I:GOS.1300
1780  READY:IFI*A>999T.P.A.Y,:IN.K:GOS.960
1790  P.A.832,"ANY ZEROS MUST";
1800  P.A.897,"BE ENTERED";:GOS.1300
1810  P.A.832,"                ";
1820  P.A.897,"          ";
1830  P.A.0," ":P.A.64," "
1840  M=100*E+10*D+C:L=100*G+10*H+I
1845  P.A.163,"AND  R:";M;
1850  IF(L*A+M)<>BT.1870
1860  P.A.68,"WELL DONE!":O=2:R=R+1:G.1880
1870  P.A.2,"SORRY- THERE'S A MISTAKE SOMEWHERE!":O=1
1880  P.A.960,"";:IN."PRESS <ENTER> TO CONTINUE ";A$
1885  IFR=10T.800
1890  REST.:CLS:G.200
2000  D.272,274,276,279,281,283,285,287
2010  D.155,347,345,343
2020  D.475,473,471,477
2030  D.157,541,539,537,535
2040  D.669,667,665,671
2050  D.159,735,733,731,729
2060  D.863,861,859
```

****** L1/4K STARSHOOT ******

TRS-80/SYSTEM-80

```
100  REM    ******** STAR SHOOT   L1/4K ***************
200  REM (C) M.S.YOUNG  MAY 82
300  REM
400  REM
500  CLS:Y=1:N=0:T=0:A$="              "
510  FOR I=1 TO 9:READ A(I+10):A(I)=-1:NEXT I
520  A(5)=1:X=1:Z=0 : INPUT"DO YOU WANT INSTRUCTIONS (Y/N)";L
522  IF L=1 THEN 3000
524  GOTO 4000
526  GOSUB 2000
530  PRINT AT 788,"SHOOT AT ";:INPUT S:PRINT AT 660,A$;A$;
540  IF (S<1)+(S>9)PRINT AT660,"INVALID SQUARE";:GOTO 530
570 IFA(S)=-1PRINT AT660,"YOU CAN ONLY SHOOT STARS";:GOTO530
580  ON S GOSUB 1001,1002,1003,1004,1005,1006,1007,1008,1009
590  T=T+1
600  IF E=-9 PRINT AT660,"NO STARS LEFT  !!!";:GOTO 640
620  IF (E<>7)+(A(5)=1) THEN 530
630  PRINT AT 660,"CONGRATULATIONS YOU TOOK";T;"MOVES"
640  PRINT AT 788,"GAME OVER PLAY AGAIN (Y/N)";:INPUT L: IF L <>
1 END
650  RESTORE :GOTO 500
1001  A(1)=-A(1):A(2)=-A(2):A(4)=-A(4):A(5)=-A(5):GOTO2000
1002  A(1)=-A(1):A(2)=-A(2):A(3)=-A(3):GOTO2000
1003  A(2)=-A(2):A(3)=-A(3):A(5)=-A(5):A(6)=-A(6):GOTO2000
1004  A(1)=-A(1):A(4)=-A(4):A(7)=-A(7):GOTO2000
1005  A(2)=-A(2):A(4)=-A(4):A(5)=-A(5):A(6)=-A(6):A(8)=-A(8):GOT
02000
1006  A(3)=-A(3):A(6)=-A(6):A(9)=-A(9):GOTO2000
1007  A(4)=-A(4):A(5)=-A(5):A(7)=-A(7):A(8)=-A(8):GOTO2000
1008  A(7)=-A(7):A(8)=-A(8):A(9)=-A(9):GOTO2000
1009  A(5)=-A(5):A(6)=-A(6):A(8)=-A(8):A(9)=-A(9):GOTO2000
2000  REM PRINT BOARD AND TEST FOR GAME END
2010  E=0 : FOR I=1 TO 9
2020  IF A(I)*X=1 THEN 2040
2030  PRINT AT A(I+10)+Z,".";:GOTO 2050
2040  PRINT AT A(I+10)+Z,"*";
2050  E=E+A(I) : NEXT I : RETURN
3000  REM INSTRUCTIONS
3010 P.A.0,"THE OBJECT OF THE GAME IS TO SHOOT STARS ON A BOARD"
3020  FOR I=1 TO 9 : PRINT AT A(I+10)+25,I;:NEXT I
3030 P.A.64,"NUMBERED AS IN THE CENTER DIAGRAM TO GET A PATTERN"
3040 P."AS SHOWN IN THE LAST DIAGRAM FROM THE FIRST"
3050  X=1:Z=0:GOSUB 2000:X=-1:Z=53:GOSUB 2000
3060 P.A.192,"TO SHOOT A STAR ENTER THE NUMBER OF IT'S POSITION"
3070 P.:P."WHEN A STAR IS SHOT IT WILL EXPLODE CHANGING THE PATT
ERN"
3080  P."THE AREA AFFECTED DEPENDS ON WHICH STAR IS SHOT "
3085  P."A STAR WILL CHANGE TO A DOT AND A DOT TO A STAR"
3090  INPUT "PRESS ENTER FOR MORE INSTRUCTIONS";B$
3100  P.A.0,"A SHOT IN THE CORNER WILL CHANGE THE 4 IN THAT CORN
ER"
3110 P."A SHOT TO POSITION 1 WILL CHANGE POSITIONS 1,2,4,5"
3115 P.
3120  P."A SHOT IN THE CENTER OF A SIDE WILL CHANGE ALL THAT SID
E"
3130  P."A SHOT TO POSITION 2  WILL CHANGE POITIONS 1,2,3"
3135 P.
3140 P."A SHOT IN THE CENTER WILL CHANGE ALL BUT THE CORNERS"
3150 P."A SHOT TO POSITION 5 WILL CHANGE POSITIONS 2,4,5,6,8"
3160  P.:INPUT "PRESS ENTER TO START GAME";B$
4000  CLS
4010 X=1:Z=-640:GOSUB 2000
4020 FOR I=1 TO 9 :PRINT AT A(I+10)-615,I;:NEXT I
4030  X=-1:Z=-587:GOSUB 2000
4040  P.A.384,"   START";TAB(28);"LAYOUT";TAB(55);"FINISH"
4050 P.:P."YOU CAN GET FOR START TO FINISH IN 11 MOVES GOOD LUCK
 !!"
4060  X=1:Z=0:GOTO 526
9999  DATA640,645,650,768,773,778,896,901,906
```

***** INDEX TO VOLUME 3 .. DECEMBER 1981 TO JULY 1983 *****
(Volume 3 includes Dec. 81, Jan-Oct 82, & July 83)

SOFTWARE PEACH/COLOUR COMPUTER			PAGE
ATOMIC TABLES	P/CC	Apr 82	20,26,28
BIORHYTHMS	P/CC	Oct 82	21,26,28
CALENDAR	P/CC	Sep 82	15,21,25
CHEQUE BOOK DATA FILE	P/CC	Aug 82	16,25,27
CUBE (3D)	P/CC	May 82	20,25,26
HANGMAN	CC	Sep 82	15,22
INCOME TAX CALCULATOR	P/CC	May 82	19,24,25
LOAN CALCULATION	P/CC	Oct 82	21,26,27
MASTERMIND	P/CC	Jul 83	15,22,25
MILEAGE CALCULATOR	P/CC	Jul 82	18,25,26
MULTIPLE REGRESSION ANALYSIS	P/CC	Apr 82	20,27
NORMAL DISTRIBUTION	P/CC	Jun 82	17,24
PAYROLL	P/CC	Aug 82	17,25,28
PRINTED DIRECTORY LISTING	P/CC	Oct 82	3
SECTOR EDITOR	P	Sep 82	16,23
SINK THE ENEMY NAVY	P/CC	Jul 83	14,19,22
3-D CUBE	P/CC	May 82	20,25,26
UNIT CONVERSIONS (METRIC)	P/CC	Jun 82	16,21,22
VARIABLE WORKSHEET	P/CC	Jul 82	18,24,26

SOFTWARE - LEVEL 1			
CLEANUP	L1/4k	Jul 83	17,28
MATRIX MANIPULATION	L1/4k	Oct 82	23,32
SPACE COMMANDER	L1/4k	Apr 82	20,30
SPACE GALAXY	L1/4k	Apr 82	21,31
TRIANGLE SOLUTIONS	L1/4k	Dec 81	9
WHEEL LOADER PRODUCTION	L1/4k	Mar 82	23,30

SOFTWARE - LEVEL 2			
ALIEN INVASION	L2/16k	Mar 82	22,26
ANAGRAMS	L2/16k	Jul 83	17,29
BASIC + LABELS	DB/16+k	Dec 81	22
+ MICROBUG		Jan 82	35
CALENDAR	L2/16k	Jul 82	18,28
CHECKSUM	L2/4kml	Oct 82	22,31
CHEQUE ACCOUNT MANAGER	L2/16k	Aug 82	20,32
CODE BREAKER	L2/4k	Oct 82	25,34
COMPOSER - MICROBUG		Oct 82	20
CRICKET	L2/16k	Sep 82	20,27
CURVILINEAR REGRESSION ANALYSIS	L2/16k	Dec 81	28
DATA BASE MGT. SYSTEM	DB/48k	Jan 82	15,29
DRAW (IN BASIC BASIC)*	L2/16k	Apr 82	15
DR. WHO ADVENTURE	L2/16k	Jul 82	21,31
+ INPUT/OUPUT		Oct 82	5,6
+ MICROBUG		Jul 83	14
DUPLEX	L2/ml	Jan 82	16,31
E=MC SQUARED	L2/4k	Jul 83	17,29
FASTER CASSETTE ROUTINES	L2/16kml	Sep 82	18,29
FAULT FINDER	L2/16k	Aug 82	19,32
FILES	L2/48k	Jan 82	13,27
FLASHING MESSAGE ROUTINE	L2/16k	Feb 82	23,29
FLEXITIME	L2/4k	Oct 82	25,33
FLIP	L2/16k	Apr 82	23,32
GOLF	L2/16k	Jul 83	15,25
GRAPHIC PACKER (AUTOMATIC)		Jul 82	15-17
HEX CONSTANTS	L2/16k	Jul 82	19,29
HORSE PERF. GUIDE (SA)	L2/16k	Jul 83	17,31
INCOME TAX CALCULATOR	L2/16k	Feb 82	27,34
INPUT DEMONSTRATION	L2/4k	Oct 82	24,33
JUMBLED PLAYERS	L2/16k	May 82	21,31
JUMP THE RAPIDS	L2/16kml	Aug 82	19,30
+ MODEL 3 MICROBUG		Oct 82	20
KEYWORD UTILITY 1.5	L2	Apr 82	23,33
LABELS (BASIC +)	DB/16+k	Dec 81	22
LEMNISCATES	L2/4k	Mar 82	25,34
LIST DISABLE	*L2/m.l.	Feb 82	6
LOAN CALCULATION	L2/16k	Jun 82	20,33
LONGVARS	L2/48k	Feb 82	25,31
LOWER CASE CONVERTER	L2/16kml	Jul 82	21
MAGIC CUBE SOLVER	L2/16k	May 82	23,29
MATRIX MANIPULATION	L2/16k	May 82	21,33
MAZE (3D)	L2/16k	Dec 81	13
MERGE PROGRAM IN 1 LINE	*L2/4k	Sep 82	5
MICRO GRAND PRIX	L2/16k	Jun 82	17,25
+ MICROBUG		Aug 82	8
MIND READER	L2/4k	Feb 82	27,33
MORSE PRACTICE	L2/16kml	Aug 82	22,24
MOVE BY 1's	L2/16k	Oct 82	22,29
MOVIE UTILITY FOR MOD. 3 MICROBUG		Oct 82	20
OTHELLO	L2/16k	Jun 82	20,31
PASSWORD	L2/16k	Jun 82	19,29
POLYNOMIAL REGRESSION ANALYSIS	L2/16k	Dec 81	26
POTHOLE	L2/16k	Sep 82	20,28
PRINTER DRIVER SCRIPSIT/SERIAL	*L2/ml	Sep 82	5-7
PROPERTY INVESTMENT SPECULATION	L2/4k	Sep 82	17,26
PUNCTUATION	L2/16k	Mar 82	25,34
QUICKSORT (IN SORTING ARTICLES)BASIC/ml/BASIC DRIVER		Jul 82	11-14
READ-A-LINE	L2/16k	Apr 82	21,32
RESET	DB/32k	May 82	22,34
RESTORE (LINE NO.) INCL SA/NT POSTCODES	L2/16k	Mar 82	23,30
S.A. HORSE PERFORMANCE GUIDE	L2/16k	Jul 83	17,31
SAVER AND LOADER	*L2/ESF	Jun 82	15
SCREEN COPY UTILITY	L2/ml	Feb 82	23,31
SCREEN FORMATTING NOS.	*	Aug 82	7
SERIES IMPEDANCE CIRCUIT	L2/16k	Jul 82	20,29
SKYDIVER	L2/16k	May 82	20,26
SOLITAIRE PATCH	L2/16k	Mar 82	25,33
SORTING - SEE ARTICLES	*Jan,Feb,Mar,Jul 82,Jul83		
SOUND GENERATION	*L2&ml	Feb 82	9
STEEPLECHASE	L2/16k	Dec 81	30
SUPER HANGMAN	L2/16k	Jan 82	13,24
SYSTEM 80 m/l PATCHES	*L2	Jun 82	5-7
3-D MAZE	L2/16k	Dec 81	13
VARIABLE LISTER	*L2	Jun 82	8

* = PROGRAM IS PART OF AN ARTICLE, AND NOT OTHERWISE INDEXED.

HARDWARE		
ADDING A JOYSTICK	Sep 82	13
COMPUTER ANATOMY - PART 1	Oct 82	11
DOUBLE SYSTEM-80 RAM FOR $15	Jul 83	10
EXPANSION INTERFACE FUNCTIONS	Feb 82	22
JOYSTICKS & I/O PORTS PART 5	Jan 82	4
JOYSTICKS & I/O EDGE CONNECTORS	Jul 82	7
MICRO-80 L/CASE FOR SYSTEM-80	May 82	14
SIMPLE INTERFACE FOR MODEL 33 TELETYPE	Mar 82	9
SM ELECTRONICS FOR OLYMPIA ES100	May 82	16
SYSTEM-80 CLOCK MODIFICATION	Aug 82	10
+ MICROBUG	Oct 82	20
SYSTEM-80 ADD-ON KEYPAD	Aug 82	15
VIDEO CONTRAST IMPROVED -CHIFFON	Sep 82	5
WINCHESTER DISK INTRO. -EDITORIAL	Feb 82	2

MICROBUGS

BASIC + LABELS	Jan 82	35
COMPOSER/BAS CHANGES	Oct 82	20
DEF FN ADDENDUM	Oct 82	19
DOCTOR WHO ADVENTURE	Jul 83	14
FREE SOFTWARE LIBRARY - COMPOSER	Jul 83	14
HOUSEHOLD ACCOUNTS	Sep 82	8
JOYSTICKS & I/O PORTS - EDGE CONNECTOR STANDARDS	Jul 82	7
JUMP THE RAPIDS ON MODEL 3	Oct 82	20
LEVEL 1 IN LEVEL 2	Sep 82	7
LOTTO PREDICTOR (SEP 80)	Jul 82	6
MICRO GRAND PRIX	Aug 82	8
MOVIE UTILITY FOR MODEL 3	Oct 82	20
SAVING/LOADING m/l PROGS ON WAFER	Jul 82	6
SYSTEM-80 CLOCK MODIFICATIONS	Oct 82	20

EVALUATIONS AND REVIEWS - SOFTWARE

ACCEL3 vs. ZBASIC	Oct 82	13
BATTLEGROUND	Sep 82	11
BRYANT'S EDUCATIONAL - MONEY BOX - DECIMAL DAN - ALADDIN - TOWERS & CROCODILE	Apr 82	7
DATESTONES OF RYN (DUNJONQUEST)	Sep 82	9
DUNJONQUEST - TEMPLE OF APSHAI	May 82	12
DUPLITAPE	Sep 82	11
+ AVAILABILITY	Oct 82	13
ESOS - EXATRON FLOPPY OPER. SYS	Aug 82	13
FS1 - FLIGHT SIMULATOR	May 82	13
MICROSOFT LEVEL 3 BASIC	Oct 82	17
MODEL 4 TRS-80 (EDITORIAL)	Jul 83	2
TC-8 CASSETTE OPERATING SYSTEM	Sep 82	12
TRS-80 ASTROLOGY	Jul 83	13
VERSAFILE	Jul 82	7
VISICALC	Sep 82	10
WORP-1 DICK SMITH'S WORD PROCESSOR	Oct 82	18
ZBASIC vs. ACCEL3	oct 82	13

EVALUATIONS AND REVIEWS - HARDWARE

LNW-80 MARK 2 COMPUTER	Oct 82	15
TRS-80 COLOUR COMPUTER	Dec 81	3
TRS-80 COLOUR COMPUTER	Apr 82	17

INPUT/OUTPUT (LETTERS TO EDITOR)

Those with helpful information only

ASYLUM ADVENTURE HINTS	Oct 82	6,7
BLOCK MOVING MORSE PROGRAM	Oct 82	6
BREAK DISABLE	Jan 82	5
COPYING MICROCHESS BY COPIER	Jun 82	4
CURSOR MEMORY POSITIONS	Jun 82	3-4
DATA LOST FROM TAPES	Jul 82	4
DR. WHO ADVENTURE - ENOUGH MEMORY	Oct 82	6
DR. WHO ADVENTURE - ENOUGH MEMORY	Jul 83	3
ESCAPE FROM TRAMM SOLUTION	Jul 82	4
ESCAPE TO TRAMM	Jun 82	4
FILES FOR SYSTEM-80	May 82	17
LOADING EDTASM+ SYSTEM TAPES/ RECORDER HEAD ALIGNMENT	Jul 83	4
LOCKING OUT RESET IN LEVEL 2	Sep 82	3
MAKING SYSTEM TAPE FROM TANDY MICROCHESS TAPE	Apr 82	4
NON-USE OF DEFFN IN SYSTEM-80	Sep 82	3
ONE LINE MERGE PROGRAM	Sep 82	4
POKING m/l INTO MEMORY	Jul 82	3
PRINTER DRIVER - SCRIPSIT/SERIAL	Sep 82	5-7
REAL TIME CLOCK LOCATION/MEMORY	Aug 82	4
RELOCATABLE CALLS IN m/l	Aug 82	4
REVERSING PRINT & LPRINT ON SYSTEM-80	Apr 82	20
SARGON ON DISK	Mar 82	2
SAVING OBJECT FILES ON ESF	Mar 82	3
SCRIPSIT CONVERSION FOR SYSTEM-80	May 82	16
SKYDIVER ERRORS/COMPUTER FAULT	Jul 82	5
SLOW LIST FUNCTION	Jan 82	6
SOUND EFFECTS RE-VISITED	Feb 82	4
SYSTEM-80 - COPYING TAPE TO DISK	Jul 82	4
SYSTEM-80 - MODS FOR JOYSTICK	Sep 82	3
SYSTEM-80 - SOUND EFFECTS RE-VISIT	Jul 82	5
SYSTEM-80 - TAPE HEAD CLEANING	Jul 82	4
TABBING BEYOND 64	Jan 82	5
VIDEO CONTRAST IMPROVEMENT USING CHIFFON	Sep 82	5

MISCELLANEOUS ARTICLES AND ITEMS

BASIC BASIC PART 1 (INCL DRAW) #	Apr 82	13
BASIC BASIC PART 2	Jun 82	7
BASIC INTERCHANGE BETWEEN 80's AND OTHER COMPUTERS	Jun 82	10
BASIC RECURSION	APR 82	10
BETTER BASIC PROGRAMMING PART 7 #	Feb 82	6
ENCHANCEMENTS TO MODEL 2 (MODEL 16)	Dec 81	2
INDEX TO VOLUME 2	Dec 81	6-7
IT ALL DEPENDS (LOGICAL OPERATORS IN CONDITIONAL STATEMENTS)	Mar 82	4
LESS BASIC BASIC	Aug 82	7
LEV 1 - LEV 2 PROGRAM CONVERSION	Apr 82	5
LOWER CASE IN NEW MODEL 1	Dec 81	2
MAKING DECISIONS THE EASY WAY	Oct 82	9
MODEL 3 GETS RUN-AROUND AS STATUS SYMBOL	May 82	9
NEWDOS 2.1 - JKL FEATURE v. MX-80	May 82	18
PRINCIPLES & GUIDELINES FOR WRITING INSTRUCTIONAL PROGRAMS	May 82	4
SAVE & LOAD m/l PROGS ON WAFER	Jun 82	13
+ MICROBUG	Jul 82	6
SAVING DISK SPACE BY MERGING PROGRAMS OR ROUTINES	#Sep 82	8
SUBMITTING PROGS FOR PUBLICATION	Oct 82	7
THEORY + TECH. FOR SORTING - 3	#Jan 82	6
+ MICROBUG	#Feb 82	5
THEORY + TECH. FOR SORTING - 4	#Feb 82	10
THEORY + TECH. FOR SORTING - 5	#Mar 82	12
THEORY + TECH. FOR SORTING - 6	#Jul 82	8
THEORY + TECH. FOR SORTING - 7	#Jul 83	6
TRANSFERRING TOUCHTYPE TO DISK	#Feb 82	21
USING EDTASM, SCRIPSIT & m/l PROGRAMS ON SYSTEM-80	Jun 82	4-7
USING THE LEV 1 ARRAY EFFECTIVELY	Feb 82	21

KALEIDOSCOPE - COLOUR COMPUTER NOTES

MEMORY POINTER LOCATIONS	Jul 83	4
PRINTING DISK DIRECTORY	Oct 82	3
SPEED UP POKES	Jul 83	4

PEACH BOWL - PEACH COMPUTER NOTES

RANDOM NUMBER DIFFERENCES	Jul 83	4
UNPROTECTING PROTECTED PROGRAMS	Oct 82	3
80x50 LINE SCREEN FORMAT	Jul 83	4

GROUP ONE - MODEL 1 NOTES

KEYBD.TIPS/DISK MOD. FOR GOLF/BAS	Jul 83	5

FORM THREE - MODEL 3 NOTES

MODEL 1 & 3 ROM DIFFERENCES	Oct 82	4
READING DISTRIBUTION DOS	Jul 83	5

***** NEXT MONTH'S ISSUE *****

Next month's issue will contain at least the following programs plus the usual features and articles. An (80) after a program title indicates that the program will be for TRS-80 Model 1/3 or System 80/Video Genie. A (CC) indicates that the program will be for the TRS-80 Colour Computer and (Peach) that the program is for the Hitachi Peach.

** DESERT CHASE (80) L2/16K **

In this graphic game of high adventure, the object is to travel across the vast Simpson Desert. Your journey is made perilous by such hazards as wild tribesmen, lack of water and sandstorms.

** FORMATION (80) L2/16K **

Formation is an Electronic Form Creation and Data Entry System. The program lets you design and store a form that will later accept operator input. The data that is entered by the operator is transformed into normal BASIC DATA lines that can be manipulated by your own program.

** OTHELLO (CC) **

Othello is written for the 16K Colour Computer, it is a game of strategy for two players and is based on the board game of the same name. The game is played on a 8x8 board and you must outflank your opponent to flip his playing pieces to your colour.

** PRIORITIES (80) L2/16K **

It seems that the Razor Gang have been responsible for a lot of things, even this program. The Doctor who wrote it needed to order the activities if his hospital into order of priority. I don't know if the razor gang will be after you, but you too can now order your priorities.

** THE TOWERS OF HANOI (CC & PEACH) **

This popular game has now been converted to run on your Hitachi Peach or Colour Computer with the addition of colour and sound. You have three pegs and a number of disks. You must get all the disks from the leftmost peg to the rightmost peg by moving one disk at a time without putting a large disk on top of a small disk. It will drive you mad!!

** REGISTER DISPLAY PROGRAM (Peach) **

This program provides an easy means of examining register contents at any time during the operation of a program. The Register Display program can provide information at any point in a program and can then revert to either the program under test, to BASIC command level or to the machine language monitor.

APPLICATION FOR PUBLICATION OF A PROGRAM IN MICRO-80

Date .....

To **MICRO-80**
SOFTWARE DEPT.
P.O. BOX 145,
MORPHETT VALE, SA., 5162

Please consider the enclosed program for publication in MICRO-80.

Name .....

Address .....

.....

.....Postcode .....

***** CHECK LIST *****

Please ensure that the cassette or disk is clearly marked with your name and address, program name(s), Memory size, Level I, II, System 1 or 2, Edtasm, System, etc. The use of REM statements with your name and address is suggested, in case the program becomes separated from the accompanying literature.

Ensure that you supply adequate instructions, notes on what the program does and how it does it, etc.

For system tapes, the start, end, and entry points, etc.

The changes or improvements that you think may improve it.

Please package securely — padabags are suggested — and enclose stamps or postage if you want your cassette or disk returned.

***** CASSETTE/DISK EDITION INDEX *****

The cassette edition of MICRO-80 contains all the applicable software listed each month, on cassette. For machine language programs copies of both the source and object file are provided. All programs are recorded twice. Level 1 programs can only be loaded into a Level 2 machine if the 'Level 1 in Level 2' program from the MICRO-80 Software Library - Vol 1 is loaded first.

Note: System 80/Video Genie computers have had different tape-counters fitted at different times. The approximate start positions shown are correct for the very early System 80 without the volume control or level meter. They are probably incorrect for later machines. The rates for a cassette subscription are printed on the inside front cover of each issue of the magazine.

The disk edition contains all applicable programs which can be executed from disk. Level 1 disk programs are saved in NEWDOS format. Users require the Level I/CMD utility supplied with NEWDOS+ or NEWDOS 80 version 1.0 to run them.

SIDE 1	TYPE	I.D.	DISK FILESPEC	APPROX. START POSITION CTR-41	CTR-80	SYSTEM 80
DEFUSR POKE	L2/4K	P	DEFPOKE/BAS	18	10	6
" "	"	"	" "	27	15	8
DEFUSR DEMO	L2/16K	D	DEFDEMO/BAS	35	19	9
" "	"	"	" "	42	23	10
DEFUSR	SYSTEM	DEFUSR	DEFUSR/CMD	49	27	12
"	"	"	" "	55	30	13
DEFUSR	EDTASM	DEFUSR	DEFUSR/EDT	62	34	15
"	"	"	" "	73	40	19
URANIUM CORE	L2/16K	C	CORE/BAS	85	47	22
" "	"	"	" "	187	104	58
MENU (MODEL 3)	EDTASM	MENU	MENU/EDT	276	154	89
" "	"	"	" "	327	183	112
MENU (MODEL 3)	SYSTEM	MENU	MENU/CMD	375	210	132
" "	"	"	" "	382	214	137
MOVIE UTILITY	SYSTEM	MOVIE		388	218	141
" "	"	"		395	222	143
SIDE 2						
STAR SHOOT	L1/4K		STARSHOT/LV1	18	10	6
" "	"		" "	77	43	21
MULTIPLICATION	L1/4K		MULTIPLY/LV1	132	74	37
" "	"		" "	198	111	63

TO:
MICRO-80, P.O. BOX 213, GOODWOOD, SOUTH AUSTRALIA. 5034.

Please RUSH to me the items shown below:

$ ..... enclosed Date .....

..... 12 month subscription to MICRO-80

..... 12 mth. subs. to MICRO-80, plus the cassette edition

..... 12 mth. subs. to MICRO-80, plus the disc edition

..... The latest issue of MICRO-80 (see inside front cover for prices)

The MICRO-80 PRODUCTS listed below:

DESCRIPTION	PRICE

TOTAL ENCLOSED WITH ORDER

☐ Cheque ☐ Bankcard ☐ Money Order

Bankcard Account Number

Signature ..... Exp. End .....

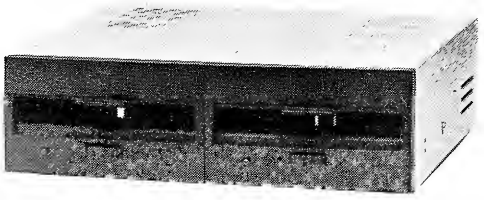
NAME .....

ADDRESS .....

Postcode .....

SAVE A PACKET ON MICRO-80's DISK DRIVE PACKAGES FOR TRS-80 MODEL 1 AND SYSTEM 80 MICROCOMPUTERS

SINGLE DRIVE PACKAGE *from* ... $499



DUAL DRIVE PACKAGE *from* ... $874

Bigger volume means lower cost price, which we are passing on to you. Avoid the annoying bundle of cables, wires and separate boxes. MICRO-80 is now offering our well-proven MPI disk drives in attractive, self-contained single or dual-drive cabinets complete with internal power supply. Our drive Ø and dual-drive packages also include the appropriate version of DOSPLUS and dual-drive cable.

The best news of all is the specially reduced package prices ...
SAVE $23 — $107 over our already low prices!

Choose the appropriate system from the table below:

DRIVE TYPE	No. of Tracks	No. of Heads	Capacity	Dosplus Version	Price	* Saving
DRIVE Ø						
1 x MPI B51	40	1	100K	3.4	$499	$137.95
1 x MPI B52	40	2	200K	3.4	$639	$97.95
1 x MPI B92	80	2	400K	3.4	$799	$107.95
DRIVE 1						
1 x MPI B51	40	1	100K	—	$415	$33.00
1 x MPI B52	40	2	200K	—	$525	$23.00
1 x MPI B92	80	2	400K	—	$695	$23.00

*Represents the saving compared with buying all the items included in the package separately

•Drive Ø package includes one bare disk drive, self-contained single-drive cabinet/power supply as illustrated, two drive cable and the version of DOSPLUS indicated.

•Drive 1 package includes one bare disk drive and self-contained single-drive cabinet/power supply as illustrated.

If it's a dual-drive system you need, then take advantage of our dual-drive package and
SAVE a further $40 on the price of two single-drive packages ...

DRIVE TYPE	No. of Tracks	No. of Heads	Capacity	Dosplus Version	Price
2 x MPI B51	40 ea	1 ea	2 x 100K	3.4	$874
2 x MPI B52	40 ea	2 ea	2 x 200K	3.4	$1125
2 x MPI B92	80 ea	2 ea	2 x 400K	3.4	$1454

Dual-drive package includes two bare disk drives, self-contained dual-drive cabinet/power supply as illustrated, two drive cables and the version of Dosplus indicated.

NOTE: All 40 track drives are completely compatible with 35 track operating systems such as TRSDOS. DOSPLUS allows you to realise an additional 14% capacity compared with TRSDOS. Under DOSPLUS 3.4, 80 track drives can read 35/40 track diskettes.

All disk drive components are still available separately:

BARE DRIVES — MPI drives offer the fastest track-to-track access time (5 milliseconds) available. All drives are capable of operating in double density for 80% greater storage capacity.

	Price	Freight
MPI B51 40 track, single-head, 100K	$349	$5.00
MPI B52 40 track, dual-head, 200K	$449	$5.00
MPI B92 80 track, dual-head, 400K	$619	$5.00
Separate, dual-drive power supply	$85	$8.00

	Price	
Self-contained, single drive cabinet/power supply	$99	$5.00
Self-contained, dual-drive cabinet/power supply	$135	$5.00
Two drive cable	$39	$2.00
Four drive cable	$49	$2.00
DOSPLUS 3.4	$149.95	$2.00

Prices are FOB Adelaide. Add $5.00 freight for single drive package, $10.00 for dual-drive package. Prices are in Australian dollars. Freight is road freight anywhere in Australia.

All items carry a 90-day parts and labour warranty. Repairs to be carried out in our Adelaide workshops.

MICRO-80

LEVEL 2 ROM

ASSEMBLY LANGUAGE TOOLKIT

by Edwin Paay

FOR TRS-80 MODEL 1, MODEL 3 AND SYSTEM 80/VIDEO GENIE

This is a new package consisting of two invaluable components:

- **A ROM REFERENCE** Manual which catalogues, describes and cross-references the useful and usable ROM routines which you can incorporate into your own machine language or BASIC programs.
- **DBUG**, a machine language disassembling debugging program to speed up the development of your own machine language programs. DBUG is distributed on a cassette and may used from disk or cassette.

Part 1 of the ROM REFERENCE manual gives detailed explanations of the processes used for arithmetical calculations, logical operations, data movements etc. It also describes the various formats used for BASIC, System and Editor/Assembly tapes. There is a special section devoted to those additional routines in the TRS-80 Model 3 ROM. This is the first time this information has been made available, anywhere. Differences between the System 80/Video Genie are also described. Part 1 is organised into subject specific tables so that you can quickly locate all the routines to carry out a given function and then choose the one which meets your requirements.

Part 2 gives detailed information about each of the routines in the order in which they appear in the ROM. It describes their functions, explains how to use them in your own machine language programs and notes the effect of each on the various Z80 registers.

Part 2 also details the contents of system RAM and shows you how to intercept BASIC routines. With this knowledge, you can add your own commands to BASIC, for instance, or position BASIC programs in high memory — the only restriction is your own imagination!

The Appendices contain sample programmes which show you how you can use the ROM routines to speed up your machine language programs and reduce the amount of code you need to write.

DBUG: Eddy Paay was not satisfied with any of the commercially available debugging programs, so he developed his own. DBUG: allows you to single-step through your program; has a disassembler which disassembles the next instruction before executing it or allows you to bypass execution and pass on through the program, disassembling as you go; displays/edits memory in Hex or ASCII; allows Register editing; has the ability to read and write System tapes and all this on the bottom 3 lines of your screen, thus freeing the rest of the screen for program displays. Four versions of DBUG are included in the package to cope with different memory sizes.

The best news of all is the price. The complete Level 2 ROM ASSEMBLY LANGUAGE TOOLKIT is only:

— Aus. $29.95 + $2.00 p&p
— UK £18.00 + £1.00 p&p

SPECIAL OFFER TO OWNERS OF THE LEVEL II ROM REFERENCE MANUAL ...
UPGRADE TO THIS ASSEMBLY LANGUAGE TOOKIT FOR ONLY $19.95!
Send back your original Level II ROM Reference Manual plus a cheque, money order or Bankcard authorisation for $19.95 plus $2.00 p&p and we will send you the new ASSEMBLY LANGUAGE TOOLKIT

MICRO-80